ARI Research Note 88-112

Design Specifications for Product To Estimate Manpower Requirements of System Designs

Eleanor Criswell, Rob Williford, and Mike Smith

Science Applications International Corp.

for

Contracting Officer's Representative
Christine R. Hartel

Manned Systems Group
John F. Hayes, Chief

Systems Research Laboratory
Robin L. Keesee, Director

September 1988

United States Army
Research Institute for the Behavioral and Social Sciences

U.S. ARMY RESEARCH INSTITUTE FOR THE BEHAVIORAL AND SOCIAL SCIENCES

A Field Operating Agency Under the Jurisdiction of the Deputy Chief of Staff for Personnel

EDGAR M. JOHNSON
Technical Director

JON W. BLADES
COL, IN
Commanding

Research accomplished under contract
for the Department of the Army

Applied Science Associates, Inc.

Technical review by

Jonathan Kaplan
John E. Stewart

NOTICES

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1a. REPORT SECURITY CLASSIFICATION Unclassified	1b. RESTRICTIVE MARKINGS - -
2a. SECURITY CLASSIFICATION AUTHORITY - - 2b. DECLASSIFICATION / DOWNGRADING SCHEDULE - -	3. DISTRIBUTION / AVAILABILITY OF REPORT Approved for public release; distribution is unlimited.
4. PERFORMING ORGANIZATION REPORT NUMBER(S) - -	5. MONITORING ORGANIZATION REPORT NUMBER(S) ARI Research Note 88-112
6a. NAME OF PERFORMING ORGANIZATION Applied Science Associates 6b. OFFICE SYMBOL (If applicable) - -	7a. NAME OF MONITORING ORGANIZATION U.S. Army Institute for the Behavioral and Social Sciences
6c. ADDRESS (City, State, and ZIP Code) P.O. Box 1072 Butler, PA 16003	7b. ADDRESS (City, State, and ZIP Code) 5001 Eisenhower Avenue Alexandria, VA 22333-5600
8a. NAME OF FUNDING / SPONSORING ORGANIZATION same as 7a. 8b. OFFICE SYMBOL (If applicable) PERI-SZ	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER MDA903-86-C-0414
8c. ADDRESS (City, State, and ZIP Code) same as 7b.	10. SOURCE OF FUNDING NUMBERS PROGRAM ELEMENT NO. 6.30.07 PROJECT NO. A792 TASK NO. (121) 1204 WORK UNIT ACCESSION NO. C2
11. TITLE (Include Security Classification) Design Specifications for Product to Estimate Manpower Requirements of System Designs	
12. PERSONAL AUTHOR(S) Eleanor Criswell, Rob Williford, and Mike Smith Science Applications International Corporation	
13a. TYPE OF REPORT Final Report 13b. TIME COVERED FROM 87/06 TO 88/01	14. DATE OF REPORT (Year, Month, Day) 1988, September 15. PAGE COUNT 139
16. SUPPLEMENTARY NOTATION Christine R. Hartel, contracting officer's representative Science Applications International Corporation subcontractor to Applied Science Associates	
17. COSATI CODES FIELD | GROUP | SUB-GROUP	18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number) Interface Design Evaluation, MANPRINT, Manpower, (SDW)

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

The U.S. Army Research Institute is developing a set of computerized aids for the evaluation of weapon system designs in terms of the manpower and personnel that they require. This report is a detailed design specification for software that assists in estimating the number of operators and maintainers required for a given weapons systems design to achieve that system's criterion performance. Specifications in the form of menu maps, data entry templates, a high level state transition diagram, leveled data flow diagrams, a structure chart, entity relationship diagrams, and entity definitions are provided for the user interface, software and data bases. Data base security and user acceptance are also discussed. The development of this design has not been funded, but the design specification may prove useful for other projects.

20. DISTRIBUTION / AVAILABILITY OF ABSTRACT ☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT. ☐ DTIC USERS	21. ABSTRACT SECURITY CLASSIFICATION Unclassified
22a. NAME OF RESPONSIBLE INDIVIDUAL Jonathan Kaplan	22b. TELEPHONE (Include Area Code) 202/274-8873 22c. OFFICE SYMBOL PERI-SM

DD Form 1473, JUN 86 Previous editions are obsolete.

SECURITY CLASSIFICATION OF THIS PAGE
UNCLASSIFIED

ACKNOWLEDGEMENTS

The authors gratefully acknowledge the contributions of three people. Steve Masterson worked on gathering the input data for the samples that will be presented at the final briefing. Coressa Robinson completed sample runs of the network precedence algorithm to calculate operator manpower. Betty Landee-Thompson prepared the section on user acceptance of Product 5.

DESIGN SPECIFICATIONS FOR PRODUCT TO ESTIMATE MANPOWER REQUIREMENTS OF SYSTEM DESIGNS

CONTENTS

LIST OF TABLES

LIST OF FIGURES

DESIGN SPECIFICATIONS FOR PRODUCT TO ESTIMATE MANPOWER REQUIREMENTS OF SYSTEM DESIGNS

INTRODUCTION

MANPRINT Methods Program Overview

The 6 MANPRINT Products. The purpose of ARI's MANPRINT Methods research program is to design and produce six automated MANPRINT decision aids. Figure 1 illustrates the six decision aids.

Products 1 to 4 are concerned with the pre-design phase of system development. These products are intended to influence system designs by identifying constraints that will affect the new system's design. Product 1 defines system requirements, including system performance criteria and reliability, availability, and maintainability requirements. Product 2 estimates the maximum crew size that will be available to man the new system, Product 3 estimates the soldier characteristics of this crew, and Product 4 focuses on likely available training for new system personnel.

Products 5 and 6 are to be used once the system design is available. These products are intended to evaluate system designs. Product 5 (the subject of this paper) determines how many operators and maintainers will be required to man the system. Product 6 will determine the characteristics of these operators and maintainers, and will identify any deficit between required and available personnel.

The logical relationship among the products is evident. Their use flows from aiding the design process to evaluating designs. Nevertheless, each product must be able to operate as independently as possible and be convenient to use. These products will help the Army insure that its soldiers will be able to operate and maintain system hardware and software in required numbers and at levels of performance that will ensure mission success.

The Three-Phase Development Effort. This effort is being conducted in three phases: concept development, detailed design specifications, and implementation. (This document is the Phase 2 final report.) In response to the request for proposals, many contractor teams developed approaches for all six products. Some teams were then selected to develop concepts for certain products; three teams were selected for each product. Phase 1 (October 1986 to June 1987) was concept development. Each team produced a narrative design document for evaluation.

The government then selected certain concepts to be further developed in Phase 2 (June 1987 to March 1988). One contractor team was selected for Products 1, 2, and 4. Two teams were selected for Products 3 and 5. All three teams were selected for Product 6. The purpose of Phase 2 is to produce a design specification document. (It is expected that down-selecting will occur at the end of Phase 2 for Products 3, 5, and 6). Given

<u>PRODUCTS</u>:	1	2	3	4	5	6
	System Performance Criteria	Maximum Crew Size	Limiting Soldier Characteristics	Probable Training Estimate	Predicted Manpower Requirements (Crew Size)	Personnel Characteristics Required
<u>PURPOSE</u>:	Identify Constraints on Designs				Evaluate Designs	

Figure 1. The Six Decision Aids in the MANPRINT Methods Program.

this document, a programmer could build the decision aid. Therefore, the Phase 2 document is unlike the usual Army Research Institute report; it contains information geared toward computer programmers.

Phase 3 (April 1988 to September 1989) will be product development. Operational decision aids will be produced. In addition, steps will be taken during Phase 3 to ensure the acceptance of the product by Army users. (The acceptance plan for Product 5 is described in this report.)

The Product 5 Concept

Product 5 is designed so that it will be accepted by Army users. This acceptance will depend on ease of use and accuracy of output. These two aspects of the Product 5 concept are described below.

Ease of Use. The Product 5 interface emphasizes consistency and places minimal memory demands on the user. Product 5 is menu driven; the menu format is consistent. Submenu and data entry form layouts are also consistent. In addition, Product 5 will incorporate vocabulary common to the other MANPRINT decision aids. Jargon will be avoided.

The Product 5 interface has been designed around a commercial off the shelf relational data base management package, R:BASE System V. This package was selected by the contractor teams as the preferred data base management package for the MANPRINT Methods decision aids. The interface and structure of Product 5 is compatiable with R:BASE System V.

Product 5 will place minimal memory demands on the user. The user will always know where he is in the menu structure through use of a location indicator on the screen. The extensive help facility will also lessen memory demands. The help facility will provide a definition of all menu items. The help facility will also provide a definition of each block on a form so that the user will know what type of entry is required. Suggested source documents advising the user where to find pertinent or better input data will be available through the help facility.

Product 5 makes the user's task easy by providing structured data entry forms and default values which need only to be modified. We plan to construct templates of performance objective conditions, functions, tasks, and times, for each system type. This will structure the user's task, provide the required information to the Product, and the user will only have to modify the template as necessary. This templating avoids completely the myriad problems that would ensue if users were required to enter free text data.

Accuracy of Output. Accuracy of output is affected by two factors: quality of input data and quality of the process by which the manpower estimates are calculated.

Figure 2 presents the relation of input and output data quality. Input data quality improves over time as system designs become more refined. Users will be advised as to the level of confidence they can place in the

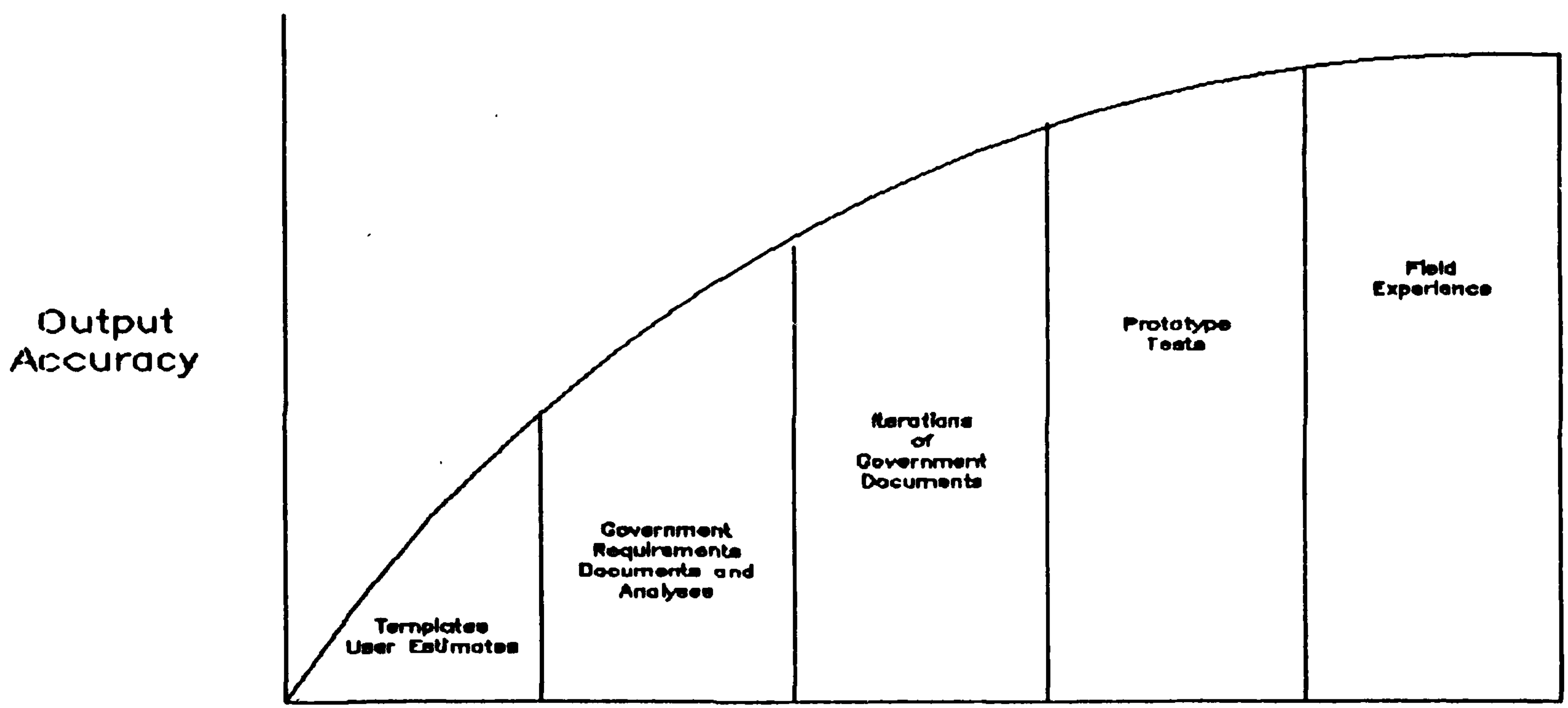


Figure 2. Product 5 Output Accuracy Relates to Input Data Quality.

accuracy of the output data, depending on the quality of their input data. Users will also be told which input data need improving.

Figure 3 presents the process by which Product 5 will estimate manpower requirements. Operator and maintainer manpower estimates are made differently.

The operator crew size calculation is based on the assumption that a job should be composed of tasks that are related to the same or similar functions, and that a person can only be one place at a time and can only do one thing at a time. Operator tasks are first grouped by function; a job is formed by using tasks within a function - this ensures that the job is built from tasks that are related. Next, the proximity relationships between functions are determined. The idea is that if a job is formed using tasks from Function X, but there is still space left in the job for more tasks, those tasks will be drawn from the next closest function. This minimizes movement for a soldier performing a job with tasks from more than one function. Product 5 assumes that a person can only be one place at a time, and that a job should contain tasks that are proximal. Next, unique jobs are formed using a standard network-precedence algorithm. This algorithm produces unique jobs, their tasks and task times, as well as an assessment of how well the job meets mission time criteria. If the design does not meet mission criteria, the user can test alternate designs until one or more is identified that appears feasible.

The maintainer crew size calculation is a straightforward multiplication of task times by yearly task frequencies divided by the number of work hours in a person year. The available data do not support the calculation of maintainer manpower using the network precedence algorithm.

Product 5 Phase 2 Design Considerations

Interface with Products 1 and 6. Product 5 is designed to be independent of the other MANPRINT products. This feature permits the Product 5 user to generate an output without having to refer to other products, which may or may not be located nearby. However, commonality in vocabulary and an understanding of how the products fit together will improve their functioning.

Product 1 generates system requirements. These requirements are stated in terms of mission, function, and task. The Kaplan and Crooks (1980) mission-function-task taxonomy was used as the basis for establishing a similar taxonomy to be used by Product 1. This taxonomy was provided to the Product 5 design team on August 3. The taxonomy is not ready for use, but an acceptable taxonomy will be developed during Phase 3. The use of some taxonomy (whether it be Kaplan and Crooks or the Product 1 taxonomy) in Product 5 is described later.

Product 5 generates the number of operator and maintainer jobs required by a system design. It lists these jobs with their associated tasks, and the criterion level at which the tasks must be performed.

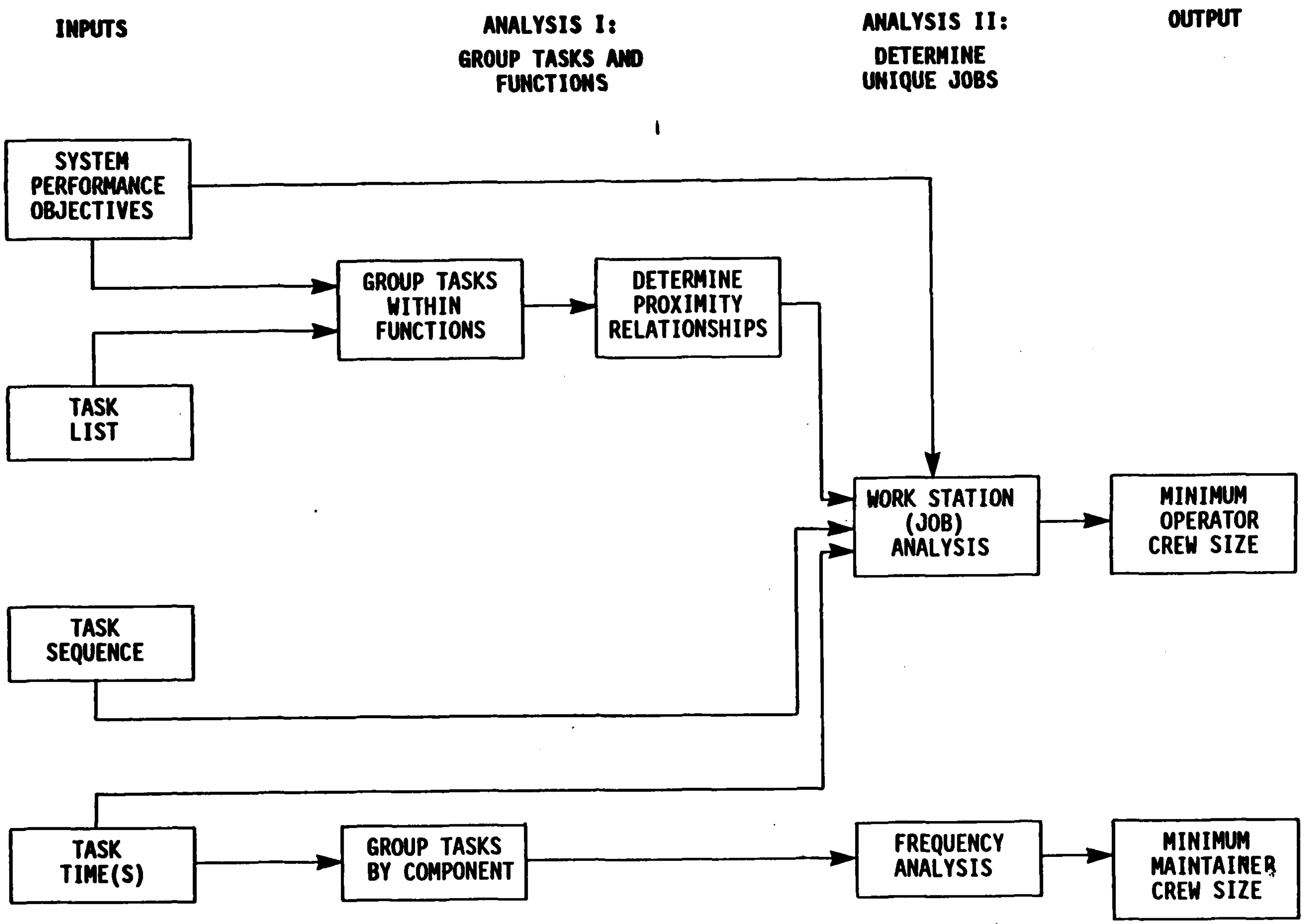


Figure 3. Product 5 Components and Process.

However, as mentioned above, the taxonomy to be used by the products is still in development. Product 6 then describes the soldier characteristics of these operators and maintainers. Product 6's analysis of Product 1 requirements and Product 5 manpower requirements allows Product 6 to state required soldier characteristics.

It is hoped that automatic communication between Products 1, 5, and 6 will save the time-consuming step of manually entering shared data. As a first step in this direction, the products will share many records in ther data dictionary. Product 1 has provided its first-cut interface specification. It is not compatible with R:BASE System V in its present form, but it will be modified significantly early in the next phase so that it can easily pass on data to the other decision aids using R:BASE.

Specifications for all MANPRINT Products. The contractor Principal Investigators and our government manager, Dr. Kaplan of ARI, have agreed on hardware, software, and interface specifications for all the MANPRINT products. These are summarized below.

All products will run on an IBM AT type computer with hard disk with at least 20 megabytes of storage. The products will be equipped with: enhanced graphics display, enhanced graphics board, 80286 processor, Bernoulli Box with two removable 20 megabyte disks, 80287 board for intensive floating point computations, 1200/2400 baud internal Hayes-compatible modem, 360 KB floppy drives, and dot matrix printer with 132 characters per inch which can emulate IBM Graphics and Epson FX/LQ series printers.

DOS 3.2 will be the operating system. Requirements for extended memory beyond 640 KB up to four megabytes will be handled via the EMS standard. R:BASE System V will be the data base management package. Microsoft C will be the programming language. Programs and data bases will be available on Bernoulli disks.

Product operation will be simple and self-evident as possible. The user will not have to memorize command language. If hierarchically nested menus more than two levels deep are used, the user must know where in the menu structure he is; the menu locator must be common across all products (commonalities across products have not been agreed upon as of this writing). The present design for Product 5 calls for two levels of menu and a deeper level of template. If a complete product run takes more than three hours, the interface must be able to return the user to last point in previous session, and inform the user which steps have been completed and which are remaining. Computer and psychology jargon should be avoided, unless the word is now in the common domain. Function key and color codes must be standard across products (to be agreed upon later).

Housekeeping procedures (e.g., closing, saving, restoring) should be common across products (to be agreed upon later). File names must be displayed so that users can select them. Select file procedures should be common across products. Editing (entering, deleting, altering, moving, and copying text) conventions should be common across products. These

conventions include keys for moving cursor, deleting, entering, and copying. These conventions should be simple and self-evident.

Users should be able to change the foreground and background colors from light to dark. Each product must include an enhanced graphics driver and printer drivers that will operate IBM Graphics and Epson FX/LQ printers.

Training will be handled by a self-evident interface and/or built-in help facility; off-line training materials will be developed only if the training need can not be handled on-line. An on-line glossary will be provided.

Approach to Product 5 Detailed Design Specification

The SAIC Product 5 team has conducted two important activities during this phase of design specification. These activities are analysis and design. The objective of analysis is to create a detailed specification of system requirements, in other words to describe what Product 5 has to provide. The object of design is to derive a solution to the problem, in other words to describe how Product 5 is to be implemented in order to satisfy the requirements detailed during analysis.

The selection of techniques for analysis and design for implementation depend upon the specific nature of the product. Traditional techniques are appropriate for Product 5, which is an information-based application. The formalization of the human interface, software, and data bases are concurrent activities and serve to complement and feed one another. The resultant specifications produced by these activities share information, but depict it in different forms. Therefore, it is important that the specifications be consistent with one another.

Consistent displays for analysis in this report have been developed for the user interface, the software, and the data bases. The user interface is expressed in menu map with data entry templates and a high level state transition diagram. The software is expressed in leveled data flow diagrams and a structure chart (deMarco, 1978; Page-Jones, 1980). The data base is expressed using entity relationship diagrams and entity definitions. Table 1 presents the three Product 5 components (human interface, software, data bases) and the techniques chosen to describe them, in both analysis and design. These displays are described below. The specifications for the data base designs are directly implementable.

Menu Map and Data Entry Templates. The menu map presents the hierarchical menu structure. Two levels of menu are used, and a deeper level of data entry templates. The menu levels and data entry templates have been developed in accordance with R:BASE System V and are presented in this report.

State Transition Diagram. State transition diagrams are useful in modeling user-product interactions. They show computer action (states), user action (operators) which enable states to change, and indicate temporal

Table 1. Product 5 Components and Analysis/Design Techniques.

Product 5 Components	TECHNIQUES Analysis	Design
Human Interface	Menu maps Data entry templates High level state transition diagram	Report templates
Software	Leveled data flow diagrams	Structure chart
Data Bases	Conceptual entity relationship diagrams Conceptual entity definitions (data dictionary)	Implementation entity relationship diagrams Implementation entity definitions

sequence with arrows as in a flow chart. We developed a high level state transition diagram for this document.

<u>Data Flow Diagrams</u>. Data flow diagrams are hierarchical graphical expressions of the exchange of information among logical data transformation objects of Product 5. Data flow diagrams consist of three symbols: circles which represent processes, parallel lines which represent data stores, and vectors which represent data flow (in the manner of DeMarco, 1978). Data flow diagrams are leveled. The highest level, Level 1, represents all of Product 5. The Level 2 and 3 diagrams expand on the most important processes (circles) in the Level 1 and 2 diagrams, respectively.

<u>Structure Chart</u>. The structure chart depicts the data flows in Product 5's primary algorithm. This is the network precedence algorithm used to create unique operator jobs.

<u>Entity Relationship Diagrams and Data Dictionary</u>. The entity relationship diagram depicts system data entities and the relationships among them. From this diagram, the entity definitions which depict entity attributes and their properties (e.g., type, precision) are developed. These are presented in tabular form in this report.

Relationship of Software and Data Base Design

The activities of software and data base analysis and design are concurrent activities. These concurrent activities serve to complement one another, and as the specifications for the two activities share data specifications (for software data stores, for data base entities), these specifications provide a means by which their consistency may be checked.

<u>Software Design Feeds Data Base Design</u>. The ability of software design to feed data base design is best described by showing that the relationship between (1) data flows and data stores of the data flow diagrams, (2) data stores and entity relationship models, and (3) data flows and entity relationship models.

The relationship between data flows and stores in the data flow diagram is a natural one. Data stores represent a time repository of data that provide for the communication of data among processes. The conventions of the methodology constrain the data store to assume the name of respective incoming/outgoing data flows.

The relationship between data stores and the information represented in entity relationship models is less direct. Data stores may represent some particular information about some data object entity, or they may represent the relationships between data object entities.

But data stores may also correspond to information that is not to be maintained in the data bases (e.g., message queues). Deriving entity models for the processes of the data flow diagram necessitates the need for a manual process during which the data stores that actually correspond to information to be maintained in the data base are identified. Further,

because data stores (individually) often represent only pieces of information about some specific data object, and (together) often reflect redundant information, data stores must be logically combined to non-redundantly reflect that information to be maintained about a data object.

There is, then, a transitive relationship between data flows and entity models. The sum of the data flows acting upon the data stores logically combined to form data object entities depicts the required user/application process transactions against the data object. These data flows represent transactions that create, delete, or use instances of the respective data object (or some subset of it), or relationships between it and other data objects. It is important to logically group and document these transactions according to data object and data object entity, because the global conceptual and implementation schemas must be specifically designed to support these transactions.

Data Base Design Feeds Software Design. Just as the activities of software design function to provide input to the data base design effort, so does the data base design activity help to feed the software design effort. The major input from data base to software design is the detailing of the composition of data objects.

As the data flow diagram is detailed along successively lower levels, software designers require more specific information about the detailed composition of data objects. In other words, at some time in the software design effort, software designers will inevitably ask "Just what information comprises data object A?" For software designers, this information is necessary to understand the processing required to manipulate the specific data object in such a manner to support that functionality required by software. Later in the design of software, data base design provides the details to perform data access functions and the interface with module logic.

Menu Map and Screen States

Levels of Menu. Product 5 uses a hierarchical menu structure. Figure 4 presents the Level 1 menu map (main menu). Figure 5 presents the Level 2 menu map.

Screen States/Data Entry Forms/Report Formats. Figures 6-12 present the data entry forms and screen states that correspond to each of the 5 Main Menu options used by Product 5. The screens meet R:BASE System V specifications. A summary of these specifications is presented in Appendix A, along with sample illustrations from the R:BASE System V manuals. Briefly, the screens support 1:1 and 1:Many relationships. The screens have two parts, the top part is the master (the 1 in the relationships), and the bottom part is the detail (the many in the relationships). The reader should note that the screens included in this report represent straightforward R:BASE designs. It is possible with R:BASE, however, to employ other interfaces. These will be studied as necessary, during Phase 3.

Figure 6 presents the main menu screen. Figures 7 and 8 corresponds to Main Menu 1: Enter/Edit System Description. Figures 7 and 8 present data entry forms for operator and maintainer manpower calculations, respectively.

Figure 9 corresponds to Main Menu 2: Generate Manpower Estimates. Screen states are shown for generating operator and maintainer manpower estimates.

Figure 10 corresponds to Main Menu 3: Generate/Print Reports. Reports are first generated, then they may be saved to a file. Subsequently, the report may be printed from the file and not generated again. A report is available for each data entry form and overall manpower estimate. There are also convenience options to allow a user to request all forms available for operators and maintainers for a given system.

Figure 11 presents the Training Menu which is Main Menu Option 4. Product 5 will include seven units of training, including four basic units and three advanced units. The lessons will be written during Phase 3, but will follow the scheme described below.

For each unit, specific instructional objectives will be identified. The instructional strategy used in addressing these objectives assumes: students must have frequent practice in the objective under study; student mastery progresses from knowledge about the concept, to a beginning-level application, to advanced level application; student progress should be measured from before to after training; student progress should be acknowledged with a certificate. The following is the general progression of each embedded training lesson.

1. Title screen

PRODUCT 5 MAIN MENU

Enter/Edit System Description	Generate Manpower Estimate	Generate/ Print Reports	Training	Data Base Maintenance

Figure 4. Menu Map, Level 1.

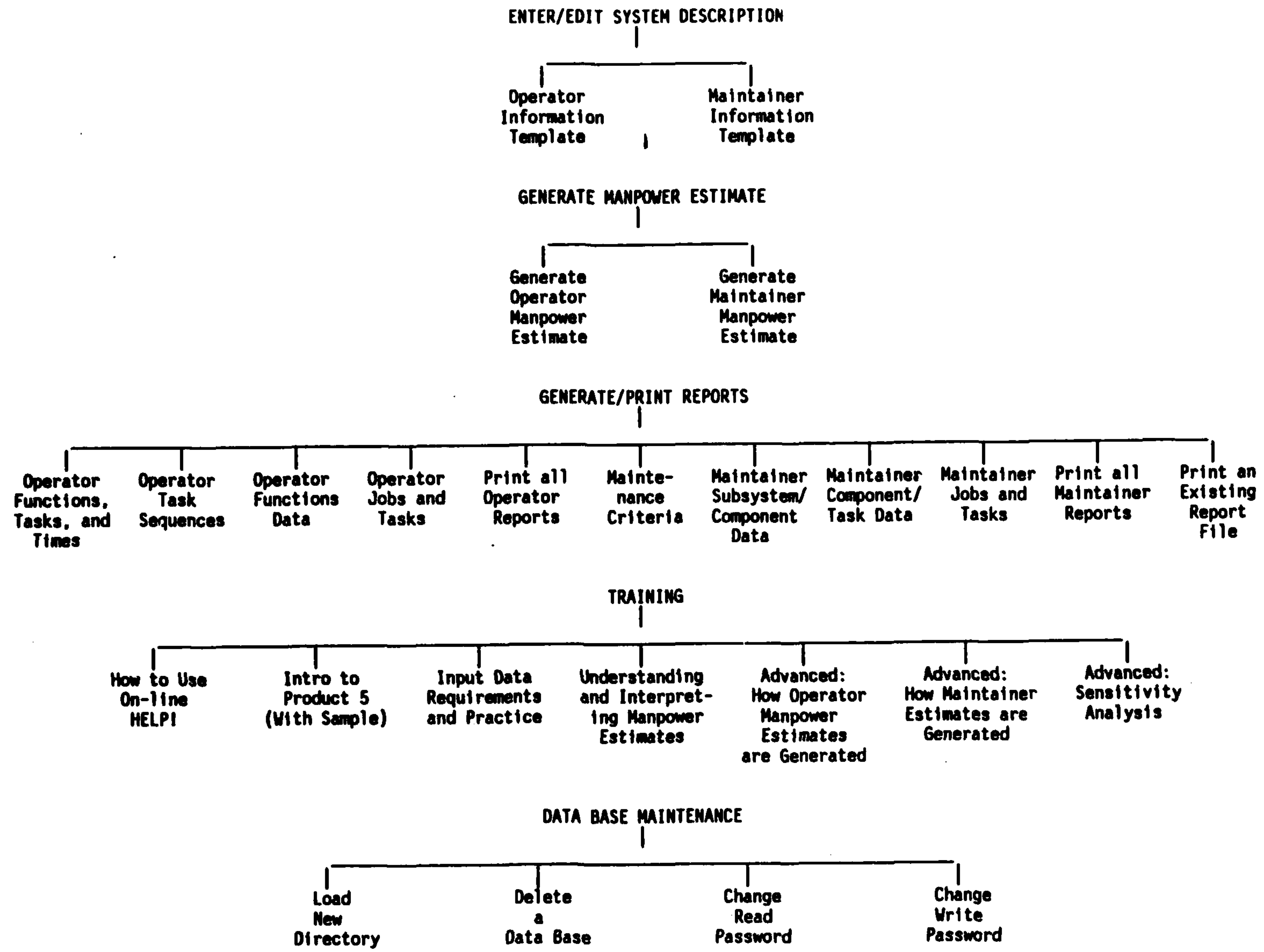


Figure 5. Menu Map, Level 2.

```
MANPRINT Manpower Estimation Aid Main Menu

        (1) Enter/edit a system description
        (2) Generate a manpower estimate
        (3) Generate/print reports
        (4) Training
        (5) Database maintenance
        (6) Exit

Enter USER password:  xxxxx

Type the number of your choice and press ENTER.
Or use arrow keys, tab key or space bar to highlight number in the menu, and
then press ENTER.
Press F10 for HELP.
---------------------------------------------------------------------------
```

Figure 6. Main Menu.

```
MENU 1 Form 1 Specify Source/Destination Data Base

Enter data base name:  xxxxx
```

```
That data base does not exist.  You will need to create one by modifying the
default-system data base as necessary.

Specify read password:   xxxxx
Specify write password:  xxxxx

                OR

The data base for your system exists.
Date of last session with that data base was xxxxx.
(ESC) Done     (FX) Data base directory     (F10) Help     (Shift F10) More
```

Figure 7. Option 1: Operator Input Data.

```
MENU 1 Form 2 Specify System Type and Taxonomy

Enter system type:  xxxxx

Enter system name:  xxxxx

(Only for new data bases)  This decision aid is designed with a standard
system-function-task taxonomy.  We strongly advise you to use the
taxonomy, rather than delete the decision aid's taxonomy and type in your
own.  (Of course, you may need to enter an occasional important function
or task name if you do not find it listed, or delete functions and tasks
that do not apply).

Do you wish to use the standard terms and taxonomy?    xxxxx

(ESC) Return      (FX) System type directory     (F10) Help
```

Figure 7 (Continued). Option 1: Operator Input Data.

```
╔══════Menu 1.1  Enter/Edit Operator or Maintainer Information══════╗
║                                                                   ║
║        (1) Enter/edit operator information                        ║
║        (2) Enter/edit maintainer information                      ║
║        (3) Exit                                                   ║
╚═══════════════════════════════════════════════════════════════════╝
(F10)  Help
```

Figure 7 (Continued). Option 1: Operator Input Data.

```
Menu 1.1.1 Enter/Edit Operator Information

        (1) Enter/edit operator performance conditions
        (2) Enter/edit operator functions, tasks, and times
        (3) Enter/edit operator task sequences
        (4) Enter/edit operator functions data
        (5) Exit
```

Enter user password: xxxxx
Enter modify password: xxxxx

Maximum number of operators possible: xxxxx

(ESC) Done   (F10) Help

Figure 7 (Continued). Option 1: Operator Input Data.

1.1.1.1a Enter/Edit Operator Performance Conditions

```
═══════════════════════════Edit Operations═══════════════════════════
Save   AddNew   Delete   Reset   Previous   Next   Quit
```

Do you want to accept benign performance conditions? xxxxx

```
           ENVIRONMENT conditions you wish to consider
------------------------------------------------------------------
                              xxxxx
                              xxxxx
                              xxxxx
```

(F5) Reset field (F7) PrevRow (F8) NextRow (F10) Help (Shift-F10) More

Figure 7 (Continued). Option 1: Operator Input Data.*

*Enter Operations are Add, Duplicate, Edit Again, Discard, and Quit.

1.1.1.1b Enter/Edit Operator Performance Conditions

Edit Operations						
Save	AddNew	Delete	Reset	Previous	Next	Quit

TERRAIN conditions you wish to consider
xxxxx
xxxxx
xxxxx

(F5) Reset field (F7) PrevRow (F8) NextRow (F10) Help (Shift-F10) More

Figure 7 (Continued). Option 1: Operator Input Data.

1.1.1.1c Enter/Edit Operator Performance Conditions

```
═══════════════════════ Edit Operations ═══════════════════════
  Save  AddNew  Delete  Reset  Previous  Next  Quit

TARGET/THREAT conditions you wish to consider
-------------------------------------------------------------------
                         XXXXX
                         XXXXX
                         XXXXX
(F5) Reset field  (F7) PrevRow  (F8) NextRow  (F10) Help  (Shift-F10) More
```

Figure 7 (Continued). Option 1: Operator Input Data.

1.1.1.1d Enter/Edit Operator Performance Conditions

```
╔════════════════════════Edit Operations════════════════════════╗
║ Save  AddNew  Delete  Reset  Previous   Next   Quit            ║
╚════════════════════════════════════════════════════════════════╝

╔════════════════════════════════════════════════════════════════╗
║            FRIENDLY conditions you wish to consider            ║
║----------------------------------------------------------------║
║                             xxxxx                              ║
║                             xxxxx                              ║
║                             xxxxx                              ║
║                                                                ║
╚════════════════════════════════════════════════════════════════╝

(F5) Reset field  (F7) PrevRow  (F8) NextRow  (F10) Help  (Shift-F10) More
```

Figure 7 (Continued). Option 1: Operator Input Data.

1.1.1.2 Enter/Edit Operator Functions, Tasks, and Times

Edit Operations

Save AddNew Delete Reset Previous Next Quit

Function: xxxxx
Function time requirement: xxxxx

Tasks	Number of soldiers required to do this task	Task Time (Actual)
xxxx	xxxxx	xxxxx
xxxx	xxxxx	xxxxx
xxxx	xxxxx	xxxxx

(F5) Reset field (F7) PrevRow (F8) NextRow (F10) Help (Shift-F10) More

Figure 7 (Continued). Option 1: Operator Input Data.

1.1.1.3 Enter/Edit Operator Task Sequences

Edit Operations

Save AddNew Delete Reset Previous Next Quit

Function: xxxxx
Task: xxxxx

	Tasks below must be completed before?	Same soldier must do both tasks?	Different soldier must do both tasks?
Task 1	xxx	xxx	xxx
Task 2	xxx	xxx	xxx
Task 3	xxx	xxx	xxx

(F5) Reset field (F7) PrevRow (F8) NextRow (F10) Help (Shift-F10) More

Figure 7 (Continued). Option 1: Operator Input Data.

1.1.1.4a Enter/Edit Operator Functions Data

```
╔══════════════════════════Edit Operations═══════════════════════════╗
║  Save  AddNew  Delete  Reset  Previous   Next   Quit              ║
╚═════════════════════════════════════════════════════════════════════╝
Function:  xxxxx

╔═════════════════════════════════════════════════════════════════════╗
║                   Functions in order, nearest to farthest           ║
║---------------------------------------------------------------------║
║                   xxxxx                                             ║
║                   xxxxx                                             ║
║                   xxxxx                                             ║
╚═════════════════════════════════════════════════════════════════════╝

(F5) Reset field  (F7) PrevRow  (F8) NextRow  (F10) Help  (Shift-F10) More
```

Figure 7 (Continued). Option 1: Operator Input Data.

1.1.1.4b Enter/Edit Operator Functions Data

```
╔═══════════════════════════Edit Operations═══════════════════════════╗
║  Save  AddNew  Delete  Reset  Previous  Next  Quit                   ║
╚══════════════════════════════════════════════════════════════════════╝
```

Function: xxxxx
(For management/surveillance functions only): What percent of a crew member's time must be committed to this function? xxx

	Crew must be capable of performing other functions at the same time?
Function 1	xxx
Function 2	xxx
Function 3	xxx

(F5) Reset field (F7) PrevRow (F8) NextRow (F10) Help (Shift-F10) More

Figure 7 (Continued). Option 1: Operator Input Data.

```
MENU 1.1.2  Enter/Edit Maintainer Information

            (1) Enter/edit maintenance criteria
            (2) Enter/edit maintainer subsystem/component data
            (3) Enter/edit maintainer component/task data
            (4) Exit

Enter user password:  xxxxx
Enter modify password:  xxxxx

Enter maximum number of maintainers possible:  xxxx

(ESC) Done   (F10) Help    (Shift-F10) More
```

Figure 8. Option 1: Maintainer Input Data.

1.1.2.1 Enter/edit maintenance criteria

Edit Operations						
Save	AddNew	Delete	Reset	Previous	Next	Quit

Unit Maintenance: xxxxx

Intermediate Direct Support: xxxxx

Intermediate General Support: xxxxx

(F5) Reset field (F7) PrevRow (F8) NextRow (F10) Help (Shift-F10) More

Figure 8 (Continued). Option 1: Maintainer Input Data.

1.1.2.2 Enter/Edit Maintainer Subsystem/Component Data

```
Edit Operations
Save  AddNew  Delete  Reset  Previous  Next  Quit
```

Subsystem: xxxxx

System components	Quantity per application
xxxxx	xxxxx
xxxxx	xxxxx
xxxxx	xxxxx

(F5) Reset field (F7) PrevRow (F8) NextRow (F10) Help (Shift-F10) More

Figure 8 (Continued). Option 1: Maintainer Input Data.

1.1.2.3a Enter/Edit Maintainer Component/Task Data for Unit

Edit Operations

Save AddNew Delete Reset Previous Next Quit

Subsystem-component: xxxxx

Tasks	Task time (Actual)	Number of times task is performed	Per unit of measure
xxxxx	xxxxx	xxxxx	xxxxx
xxxxx	xxxxx	xxxxx	xxxxx
xxxxx	xxxxx	xxxxx	xxxxx

(F5) Reset field (F7) PrevRow (F8) NextRow (F10) Help (Shift-F10) More

Figure 8 (Continued). Option 1: Maintainer Input Data.

1.1.2.3b Enter/Edit Maintainer Component/Task Data for Direct Support

Edit Operations

Save AddNew Delete Reset Previous Next Quit

Subsystem-component: xxxxx

Tasks	Task time (Actual)	Number of times task is performed	Per unit of measure
xxxxx	xxxxx	xxxxx	xxxxx
xxxxx	xxxxx	xxxxx	xxxxx
xxxxx	xxxxx	xxxxx	xxxxx

(F5) Reset field (F7) PrevRow (F8) NextRow (F10) Help (Shift-F10) More

Figure 8 (Continued). Option 1: Maintainer Input Data.

1.1.2.3c Enter/Edit Maintainer Component/Task Data for General Support

Edit Operations						
Save	AddNew	Delete	Reset	Previous	Next	Quit

Subsystem-component: xxxxx

Tasks	Task time (Actual)	Number of times task is performed	Per unit of measure
xxxxx	xxxxx	xxxxx	xxxxx
xxxxx	xxxxx	xxxxx	xxxxx
xxxxx	xxxxx	xxxxx	xxxxx

(F5) Reset field (F7) PrevRow (F8) NextRow (F10) Help (Shift-F10) More

Figure 8 (Continued). Option 1: Maintainer Input Data.

```
MENU 2  Generate a Manpower Estimate

            (1) Generate operator manpower estimate
            (2) Generate maintainer manpower estimate
            (3) Generate operator and maintainer manpower estimates
            (4) Exit to Main Menu

(ENT) Select   (F10) Help
```

Figure 9. Option 2: Generate Manpower Estimate.

```
═══════════2.1 Generate operator Manpower Estimate═══════════
 Enter data base name:  xxxxx
```

(ESC) Done   (Fx) Data base directory (F10) Help (Shift-F10) More

Figure 9 (Continued). Option 2: Generate Manpower Estimate.

```
══════════2.2  Generate maintainer Manpower Estimate══════════

 Enter data base name:  xxxxx

 Maintenance level, unit?  xxx
 Maintenance level, direct support?  xxx
 Maintenance level, general support?  xxx
```

(ESC) Done (Fx) Data base Directory (F10) Help (Shift-F10) More

Figure 9 (Continued). Option 2: Generate Manpower Estimate.

```
═══════2.3  Generate Operator and Maintainer Manpower Estimates═══════

Enter data base name:  xxxxx

All maintenance levels?  xxx
Maintenance level, unit?  xxx
Maintenance level, direct support?  xxx
Maintenance level, general support?  xxx

(ESC) Done   (Fx) Data base directory  (F10) Help  (Shift-F10) More
```

Figure 9 (Continued). Option 2: Generate Manpower Estimate.

```
MENU 3. Generate/Print Reports

(1) Operator functions, tasks, and times
(2) Operator task sequences
(3) Operator functions data
(4) Operator jobs and tasks
(5) Print all operator reports

(6) Maintenance criteria
(7) Maintainer subsystem/component data
(8) Maintainer component/task data
(9) Maintainer jobs and tasks
(10) Print all maintainer reports

(11)  Print an existing report file
(12)  Exit to Main Menu

User password: xxxxx
Read password:  xxxxx

(ENTER) Select    (F10) Help
```

Figure 10. Option 3: Generate/Print Reports.

```
══════3.1  Operator Functions, Tasks, and Times Report══════
 Enter data base name:  xxxxx

 Specify (P)rinter, (S)creen, (F)ile:  x
 (If F) Enter file name:  xxxxx
(ESC) Return to menu   (ENTER) Select  (Fx) Data base directory    (F10) Help
```

Figure 10 (Continued). Generate/Print Reports.

3.1 OPERATOR FUNCTIONS, TASKS, AND TIMES REPORT

Date: xxxxx
Page: xxxxx

System: xxxxx System type: xxxxx
File name: xxxxx
Data base name: xxxxx

Function: xxxxx
Function time requirement: xxxxx

Tasks	Number of soldiers required to do this task	Task Time (Actual)
xxxxx	xxxxx	xxxxx
xxxxx	xxxxx	xxxxx
xxxxx	xxxxx	xxxxx

Figure 10 (Continued). Generate/Print Reports.

```
3.2 Operator Task Sequences Report
Enter data base name:  xxxxx

Specify (P)rinter, (S)creen, (F)ile:  x

(If F) Enter file name:  xxxxx
```

(ESC) Return to Menu   (ENTER) Select (Fx) Data base directory   (F10) Help

Figure 10 (Continued). Generate/Print Reports.

3.2 OPERATOR TASK SEQUENCES REPORT

Date: xxxxx
Page: xxxxx

System: xxxxx System type: xxxxx
File name: xxxxx
Data base name: xxxxx

Function: xxxxx
Task: xxxxx

	Tasks below must be completed before?	Same soldier must do both tasks?	Different soldier must do both tasks?
Task 1	xxx	xxx	xxx
Task 2	xxx	xxx	xxx
Task 3	xxx	xxx	xxx

Figure 10 (Continued). Generate/Print Reports.

```
3.3 Operator Functions Report
Enter data base name:  xxxxx
Specify (P)rinter, (S)creen, (F)ile:  x
(If F) Enter file name:  xxxxx
```

(ESC) Return to Menu   (ENTER) Select (Fx) Data base directory   (F10) Help

Figure 10 (Continued). Generate/Print Reports.

3.3 OPERATOR FUNCTIONS REPORT

Date: xxxxx
Page: xxxxx

System: xxxxx System type: xxxxx
File name: xxxxx
Data base name: xxxxx

Function: xxxxx

Functions in order, nearest to farthest

---

xxxxx
xxxxx
xxxxx

What percent of a crew member's time must be commited
to this function? xxxxx

Crew must be capable of performing
other functions at the same time?

Function 1	xxx
Function 2	xxx
Function 3	xxx

Figure 10 (Continued). Generate/Print Reports.

```
═══════════════3.4  Operator Job/Task Report═══════════
║ Enter data base name:  xxxxx                          ║
║ Specify (P)rinter, (S)creen, (F)ile:  x               ║
║ (If F) Enter file name:  xxxxx                        ║
═════════════════════════════════════════════════════════
  (ESC) Return to Menu   (ENTER) Select  (Fx) Data base directory    (F10) Help
```

Figure 10 (Continued). Generate/Print Reports.

```
3.4 OPERATOR JOB/TASK REPORT

                                                     Date:  xxxxx
                                                     Page:  xxxxx
-----------------------------------------------------------------------
System:  xxxxx          System type: xxxxx
File name:  xxxxx
Data base name:  xxxxx

Environment:            Terrain:      Target/Threat:      Friendly:
xxxxx                   xxxxx         xxxxx               xxxxx
xxxxx                   xxxxx         xxxxx               xxxxx
-----------------------------------------------------------------------
Function: xxxxx        Minimum number of jobs estimated:  xxxxx
                       Maximum manpower constraint:  xxxxx
                       Criterion time to complete function:  xxxxx
                       Actual time to complete function:  xxxxx

Task Time       Job xxxxxx   Job xxxxx    Job xxxxx    Job xxxxx
-----------------------------------------------------------------------
nnnnn           xxxxx        xxxxx        xxxxx        xxxxx
nnnnn           xxxxx        xxxxx        xxxxx        xxxxx
nnnnn           xxxxx        xxxxx        xxxxx        xxxxx
-----------------------------------------------------------------------
System:                Minimum number of jobs estimated:  xxxxx
                       Maximum manpower constraint:  xxxxx
                       Criterion time to complete all functions: xxxxx
                       Actual time to complete all functions: xxxxx
                       List of functions where criterion not met: xxxxx
                       Percent default values used:  xx

Task Time       Job xxxxxx     Job xxxxx    Job xxxxx    Job xxxxx
-----------------------------------------------------------------------
nnnnn           xxxxx          xxxxx        xxxxx        xxxxx
nnnnn           xxxxx          xxxxx        xxxxx        xxxxx
nnnnn           xxxxx          xxxxx        xxxxx        xxxxx
-----------------------------------------------------------------------
```

Figure 10 (Continued). Generate/Print Reports.

```
═══════════════3.5 All Operator Reports═══════════════
Enter data base name:  xxxxx

Specify (P)rinter, (S)creen, (F)ile:  x

(If F) Enter file name:  xxxxx
══════════════════════════════════════════════════════
 (ESC) Return to Menu   (ENTER) Select  (Fx) Data base directory    (F10) Help
```

Figure 10 (Continued). Generate/Print Reports.*

*System returns specified report 3.1 through 3.4.

```
══════════════3.6 Maintenance Criteria Report══════════════
 Enter data base name:  xxxxx

 Specify (P)rinter, (S)creen, (F)ile:  x

 (If F) Enter file name:  xxxxx
═══════════════════════════════════════════════════════════
 (ESC) Return to Menu   (ENTER) Select  (Fx) Data base directory    (F10) Help
```

Figure 10 (Continued). Generate/Print Reports.

3.6 MAINTENANCE CRITERIA REPORT

Date: xxxxx
Page: xxxxx

System: xxxxx System type: xxxxx
File name: xxxxx
Data base name: xxxxx

Unit: xxxxx

Direct support: xxxxx

General support: xxxxx

Figure 10 (Continued). Generate/Print Reports.

```
=====3.7 Maintenance Subsystem/Component Report=====
Enter data base name:  xxxxx

Specify (P)rinter, (S)creen, (F)ile:  x

(If F) Enter file name:  xxxxx
```

(ESC) Return to Menu (ENTER) Select (Fx) Data base directory (F10) Help

Figure 10 (Continued). Generate/Print Reports.

3.7 MAINTENANCE SUBSYSTEM-COMPONENT REPORT

Date: xxxxx
Page: xxxxx

System: xxxxx System type: xxxxx
File name: xxxxx
Data base name: xxxxx

Subsystem: xxxxx

System components	Quantity per application
xxxxx	xxxxx
xxxxx	xxxxx
xxxxx	xxxxx

Figure 10 (Continued). Generate/Print Reports.

```
═══════════════3.8  Maintenance Component/Task Report═══════════
║ Enter data base name:  xxxxx                                  ║
║                                                               ║
║ Do you want unit?  xxx                                        ║
║ Do you want direct support?  xxx                              ║
║ Do you want general support?  xxx                             ║
║                                                               ║
║ Specify (P)rinter, (S)creen, (F)ile:  x                       ║
║                                                               ║
║ (If F) Enter file name:  xxxxx                                ║
═════════════════════════════════════════════════════════════════
(ESC) Return to Menu    (ENTER) Select  (Fx) Data base directory    (F10) Help
```

Figure 10 (Continued). Generate/Print Reports.

3.8 MAINTENANCE COMPONENT/TASK REPORT FOR XXXXX MAINTENANCE LEVEL

Date: xxxxx
Page: xxxxx

System name: xxxxx
System type: xxxxx
File name: xxxxx
Data base name: xxxxx

Subsystem-component: xxxxx

Tasks	Task time (Actual)	Number of times task is performed	Per unit of measure
xxxxx	xxxxx	xxxxx	xxxxx
xxxxx	xxxxx	xxxxx	xxxxx
xxxxx	xxxxx	xxxxx	xxxxx

Figure 10 (Continued). Generate/Print Reports.

```
3.9 Maintenance Job/Task Report

Enter data base name:  xxxxx

Do you want unit?  xxx
Do you want direct support?  xxx
Do you want general support?  xxx

Specify (P)rinter, (S)creen, (F)ile:  x

(If F) Enter file name:  xxxxx
```

(ESC) Return to Menu   (ENTER) Select (Fx) Data base directory   (F10) Help

Figure 10 (Continued). Generate/Print Reports.

3.9 MAINTAINER JOB/TASK REPORT

Date: xxxxx
Page: xxxxx

System: xxxxx System type: xxxxx
File name: xxxxx
Data base name: xxxxx

Maintenance Level: xxxxx

Subsystem: xxxxx

Minimum number of jobs estimated: xxxxx
Maximum number of jobs constraint: xxxxx
Criterion maintenance ratio: xxxxx
Actual maintenance ratio: xxxxx

	Job xxxxx		Job xxxxx		Job xxxxx		Job xxxxx	
	Tasks	Freq	Tasks	Freq	Tasks	Freq	Tasks	Freq
	xxxxx	nnnnn	xxxxx	nnnnn	xxxxx	nnnnn	xxxxx	nnnnn
	xxxxx	nnnnn	xxxxx	nnnnn	xxxxx	nnnnn	xxxxx	nnnnn
	xxxxx	nnnnn	xxxxx	nnnnn	xxxxx	nnnnn	xxxxx	nnnnn
Time/Yr	nnnnn		nnnnn		nnnnn		nnnnn	

System:

Minimum number of jobs estimated: xxxxx
Maximum number of jobs constraint: xxxxx
Total criterion maintenance ratio: xxxxx
Total actual maintenance ratio: xxxxx

List subsystems criterion not met: xxxxx
Percent default values used: xx

Figure 10 (Continued). Generate/Print Reports.

```
═══════════3.10  All Maintenance Reports═══════════
Enter data base name:  xxxxx

Do you want unit?  xxx
Do you want direct support?  xxx
Do you want general support?  xxx

Specify (P)rinter, (S)creen, (F)ile:  x

(If F) Enter file name:  xxxxx
```
(ESC) Return to Menu   (ENTER) Select   (Fx) Data base directory   (F10) Help

Figure 10 (Continued). Generate/Print Reports.*

*System returns specified report 3.6 to 3.9.

```
════════════════3.11 Print an Existing Report File═══════
Specify file name:
Specify (P)rinter or (S)creen:  x
```

(ESC) Return to Menu   (ENTER) Select   (Fx) Data base directory   (F10) Help

Figure 10 (Continued). Generate/Print Reports.*

---

*System returns specified report 3.1 to 3.10.

```
MENU 4. Training

Select the lesson you want to use:

   (1) How to use the on-line HELP!
   (2) Introduction to MANPRINT Manpower Estimation Aid, with sample
   (3) Input data requirements and practice
   (4) Understanding and interpreting manpower estimates
   (5) Advanced:  How operator manpower estimates are generated
   (6) Advanced:  How maintainer manpower estimates are generated
   (7) Advanced:  How system design changes affect manpower
   (8) Exit to Main Menu

(ENT) Select   (F10) Help
```

Figure 11. Option 4: Training.

2. Statement of instructional objectives

3. Pretest (may be automatically scored or self-assessment type)

4. General sequence for each instructional objective:

    a. Present concept

    b. Require a student interaction

    c. Automatic evaluation of student response; branch as required

    d. Present concept/interaction/evaluation/branch sequence again as needed

    e. Require an acquisition-level application interaction, with evaluation and branching

    f. Require a generalization-level application interaction, with evaluation and branching

5. After Step 4 has been accomplished for all instructional objectives, provide mixed (e.g., concept, acquisition application, generalization application) practice with feedback over all the objectives. Three practice items are available for each objective. Evaluate and branch as required.

6. Unit posttest (automatically scored; includes two or three items per objective)

7. Print certificate of completion

Figure 12 corresponds to Main Menu Option 5: Data Base Maintenance. Screen states are shown for data base loading and deleting, and changing passwords.

State Transition Diagrams

State transition diagrams are specifically useful in modeling human-computer interactions. A sample state transition diagram for a Data Entry/Modification operation is found in Figure 13. As shown, boxes correspond to states of the computer dialogue, which are acted upon by user stimuli to transition to other computer states.

User Dialog

This section provides a brief walk-through of the user dialog. (The Product 5 team will present an example walk-through using the M109 system at the final briefing in January.)

```
MENU 5.  Data Base Maintenance

          (1)  Load new taxonomy
          (2)  Delete data base
          (3)  Change read password
          (4)  Change write password
          (5)  Exit to Main Menu

User password:  xxxxx

(ENT) Select  (Fx) Data base directory   (F10) Help
```

Figure 12. Option 5: Data Base Maintenance.

```
═══════════════════5.1  Load New Taxonomy═══════════
║ Specify directory location of files:  xxxxx      ║
═══════════════════════════════════════════════════
   (F10) Help
```

Figure 12 (Continued). Option 5: Data Base Maintenance.

```
5.2 Delete Data Base

  Specify data base name:  xxxxx

(F10) Help
```

Figure 12 (Continued). Option 5: Data Base Maintenance.

```
═══════════════════ 5.3  Change Read Password ═══════════════════
   Enter new password:  xxxxx
```
(F10) Help

Figure 12 (Continued). Option 5: Data Base Maintenance.

```
════════════════════ 5.4  Change Write Password ════════════════════
  Enter new password:  xxxxx
(F10) Help
```

Figure 12 (Continued). Option 5: Data Base Maintenance.

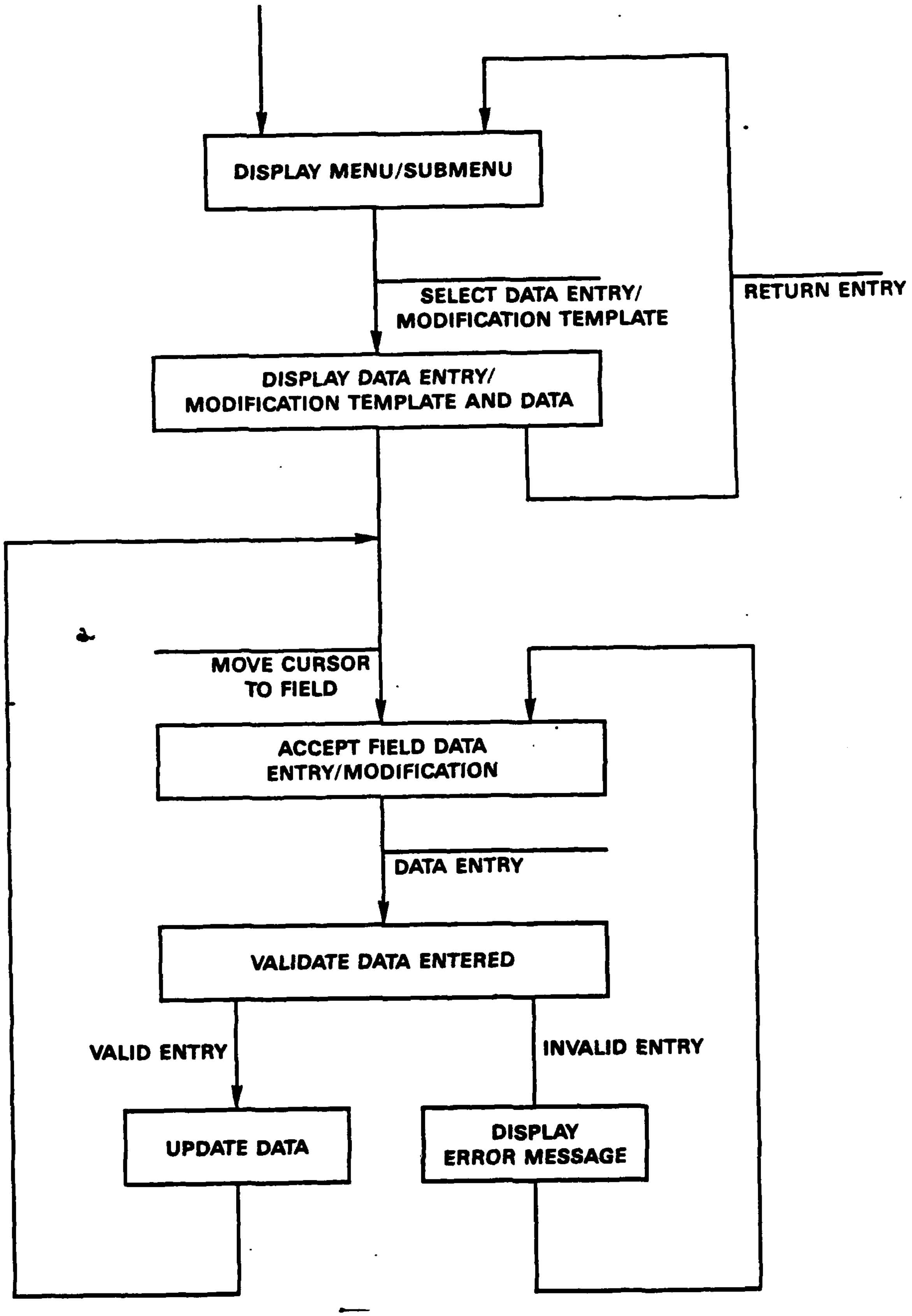


Figure 13. Sample State Transition Diagram of User Interface.

Main Menu. Upon selecting options 1, 2, or 3 (see Figure 6), but before the requested submenu is displayed, the screen is cleared, and the following prompt is given: "Enter User Password:" to which the user supplies an R:BASE USER Password.

Menu 1: Enter/Edit a System Description (refer to Figure 7). Upon selecting this option, 2 forms are displayed in sequence, after which Menu 1.1 is displayed. On Form 1, the user identifies the data base name. If data base name exists, then intent to edit existing system description is recognized, or else intent to enter a new system description is inferred. Forms are driven with R:BASE "EDIT" or "ENTER" accordingly.

On Form 2, the user identifies system type and name. The item concerning user acceptance of the standard taxonomy is displayed if "data base name" for this USER password is not found, indicating the intent is to enter rather than edit.

Form 2 includes a function key (noted on the status line) associated with a query to display list of all system types. A user is not permitted to modify the system type of a system description data base which has already been populated based on the standard taxonomy.

Menu 1.1.1: Enter/Edit Operator Manpower Calculation (refer to Figure 7). If user has accepted the standard taxonomy, then a copy of the portion of the taxonomy pertaining to the system type is made to "data base name" with OWNER password set to USER password. There is no problem if the user at a later time wishes to add maintainer information to operator information, or vice versa.

If USER password does not equal OWNER password, after the user selects an option from Menu 1.1.1 or Menu 1.1.2, but before the respective form is displayed, the user is prompted for a modify password. If the USER password does not equal modify password, then a message is displayed and the user never sees the requested form.

There are four data entry forms for operator manpower calculations: designating performance conditions; editing functions, tasks and times; determining task sequences; and, determining distance between functions.

Product 5 uses as a default benign performance conditions (Menu 1.1.1.1). The user may accept these, or select conditions he or she wishes to consider. The four categories of performance conditions shown in the screens come from the draft Product 1 conditions taxonomy. Product 5 will categorize performace conditions and combinations into three categories: low, medium, and high. Low means that the environment is not severe and performance times are shortest. Medium is a medium severe environment, and there will be some degradation, i.e., increase in task time. High is a severe environment, and task times will be even longer. We will use degradation factors developed from Siegel, Pfeiffer, Kopstein, Wilson, and Ozkaptan (1979). In addition, we will degrade task times for tasks that are susceptible to degradation, to be developed from Siegel et al. (1979).

Next the user enters/edits operator functions, tasks, and times (Menu 1.1.1.2). One function is presented per screen. The function time requirement comes either from Product 1 or a default. The tasks in the function (from the taxonomy) are listed, and the user edits the number of soldiers required to perform the task (default is "1") and the task time (default is the time associated with the latest representative of the system type). (NOTE: During Phase 3, we plan to work very closely with the Product 1 taxonomy revision effort. We would like to see the taxonomy include only one-person tasks, to the extent possible, and we would like to assure that the task list and sequence is acceptable to military experts.)

Next the user enters/edits operator task sequences (Menu 1.1.1.3). One function and one task are presented per screen. For each task, the user specifies which tasks MUST be completed before the target task can begin. The user must also specify if the same or different soldiers MUST perform this task as well as others. This information is important to the network precedence analysis. The default values will indicate that there are no constraints on either job formation as a result of task precedences or the same/different soldier question (e.g., no tasks MUST precede this one), and thus the algorithm is free to assign this task to whichever job it best fits.

Next the user enters/edits operator functions data (Menu 1.1.1.4). One function is presented per screen, and the user determines the physical proximity of that function to other functions. The user also indicates if a soldier must be assigned for management/surveillance. The user also indicates if some functions MUST be performed simultaneously; this factor affects the total operator manpower estimate, which is a result of combining the manpower estimates for each function. This question determines if the system manpower estimate is additive or can be done more economically.

Menu 1.1.2: Enter/Edit Maintainer Information (see Figure 8). The user first edits the maintenance criteria (e.g., maintenance ratios: maintenance manhours per system operating hour) for each maintenance organizational level. The defaults come from Product 1 if available, or come from previous system requirements as determined by the Product 5 team and provided in the default data base. Next, the user specifies the hardware design, by determining the system components by subsystem. The defaults come from a standard taxonomy (e.g., will be determined by the product teams during the next phase). Finally, the user determines the tasks, task times, and number of times the task is performed per unit time (e.g., per year). The default values for tasks and task times come from Maintenance Allocation charts available in Technical Manuals on representative systems. The number of times the task is performed per unit time comes from the Sample Data Collection (SDC) data base. This data base covers approximately 80 systems.

Menu 2: Generate Manpower Estimates (see Figure 9). The user indicates if he or she wants an operator, maintainer, or both manpower estimate, and enters the date base name.

Menu 3: Generate/Print Reports (see Figure 10). The user is required to enter a password to gain access to this menu option. If the USER

password does not equal the OWNER password, he or she is asked for a READ password. If the USER password does not equal the READ password, then a message is displayed and the user does not gain access to the report. If the user has access, he or she indicates the data base name is generating a report, and indicates a file name to print a previously generated report.

Menu 4: Training (see Figure 11). This item was described above. A user does not require password access to this option.

Menu 5: Data Base Maintenance. This menu option is for the system manager. Option 5.1 is to load a new taxonomy into system tables. The user specifies the directory location of the files. This action uses the R:BASE Filegateway utility. Action is not permitted if the USER password does not equal the OWNER password of the taxonomy. The owner of the taxonomy is assumed to be the system manager.

Option 5.2 is to delete a data base. The user specifies the data base name to be deleted. If the USER password does not equal the OWNER password of the specified data base name, then a message is displayed, and the data base is not deleted.

Option 5.3 and 5.4 permitted an allowed user to change the READ and WRITE password.

Help Function

The help function will have a minimum of three levels. Level 1 help is invoked by a function key. This help produces a definition of a term or procedure, with an example. Level 2 help refers to the filling in of templates. This level produces options to restart, cancel, backup, and change data before entering. Level 3 help produces the on-line glossary.

SOFTWARE ANALYSIS

Data Flow Diagrams

As mentioned in the Introduction, data flow diagrams are hierarchical graphical expressions of the exchange of information among logical data transformation objects of Product 5. (Sequence is not explicitly reflected in a data flow diagram). The diagrams are made of three symbols: circles which represent processes, boxes which represent data stores, and arrows which show data flows. Three levels of data flow diagrams are used to describe Product 5, with main process only decomposed to the third level.

Overview. Figure 14 presents the Level 1 data flow diagram for Product 5. As shown, the three high level processes of Product 5 are User Dialogue, Derive Unique Jobs, and Generate Reports. Note that the process numbering scheme reflects the hierarchies of processes.

The single external sink and source is the user, not shown, but conceptually the farthest left element. Through User Dialog, Product 5 collects data and forms three data stores. As shown, these stores are: Test system components/task function data/performance objectives/conditions; Task sequences/times/descriptions, and the Kaplan-Crooks (or whatever taxonomy is used) taxonomy.

The Derive Unique Jobs process derives input from the Task sequences/ times/description store. It provides output to the Jobs store. The process also interacts directly with User Dialogue when detecting Feasibility Errors, e.g., when the user enters constraints of time and distance which affect the construction of unique jobs.

The Generate Reports process accepts report requests from User Dialogue, extracts necessary information from various stores, generates the requested report, then either stores the report or returns it to the user (through User Dialogue) for review. Previously generated reports are returned to User Dialogue directly without processing.

User Dialogue. All the functionality of User Dialogue is provided by R:BASE. Therefore all User Dialogue software will not be written from scratch. Figure 15 presents the Level 2 data flow diagram for User Dialogue. The four processes involved in User Dialog are: Sequence Control, Data Entry/Modification, User Guidance, and Information Presentation.

Sequence Control controls the sequencing of menus/submenus, ultimately passing control to Data Entry/Modification onto Information Presentation, depending upon the user's intention. It has direct interaction with User Guidance for the display to users of help and errors related to menus and submenus.

The Data Entry/Modification process takes input from the user as shown and outputs to the three input data stores. It interacts directly with User

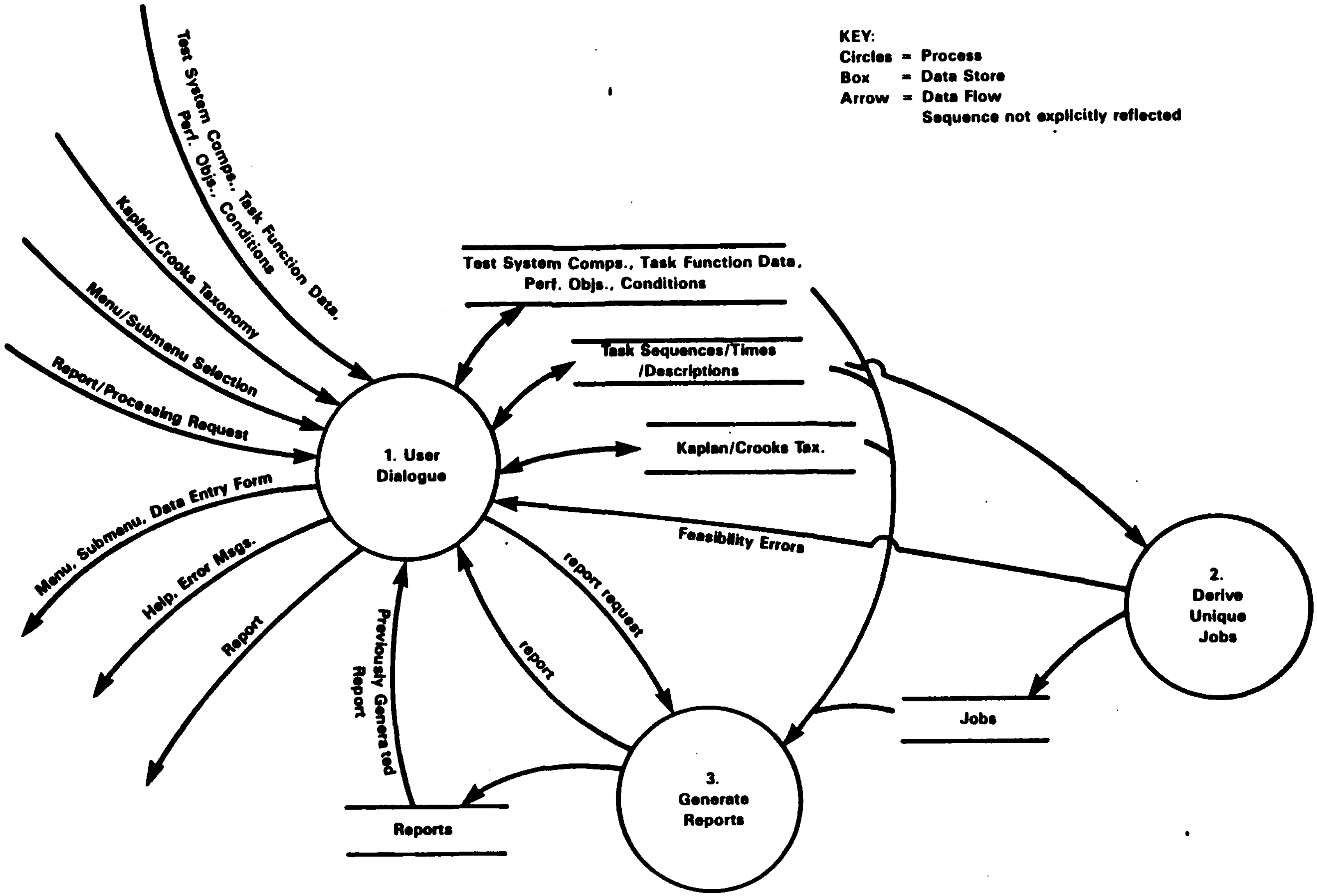


Figure 14. Level 1 Data Flow Diagram.

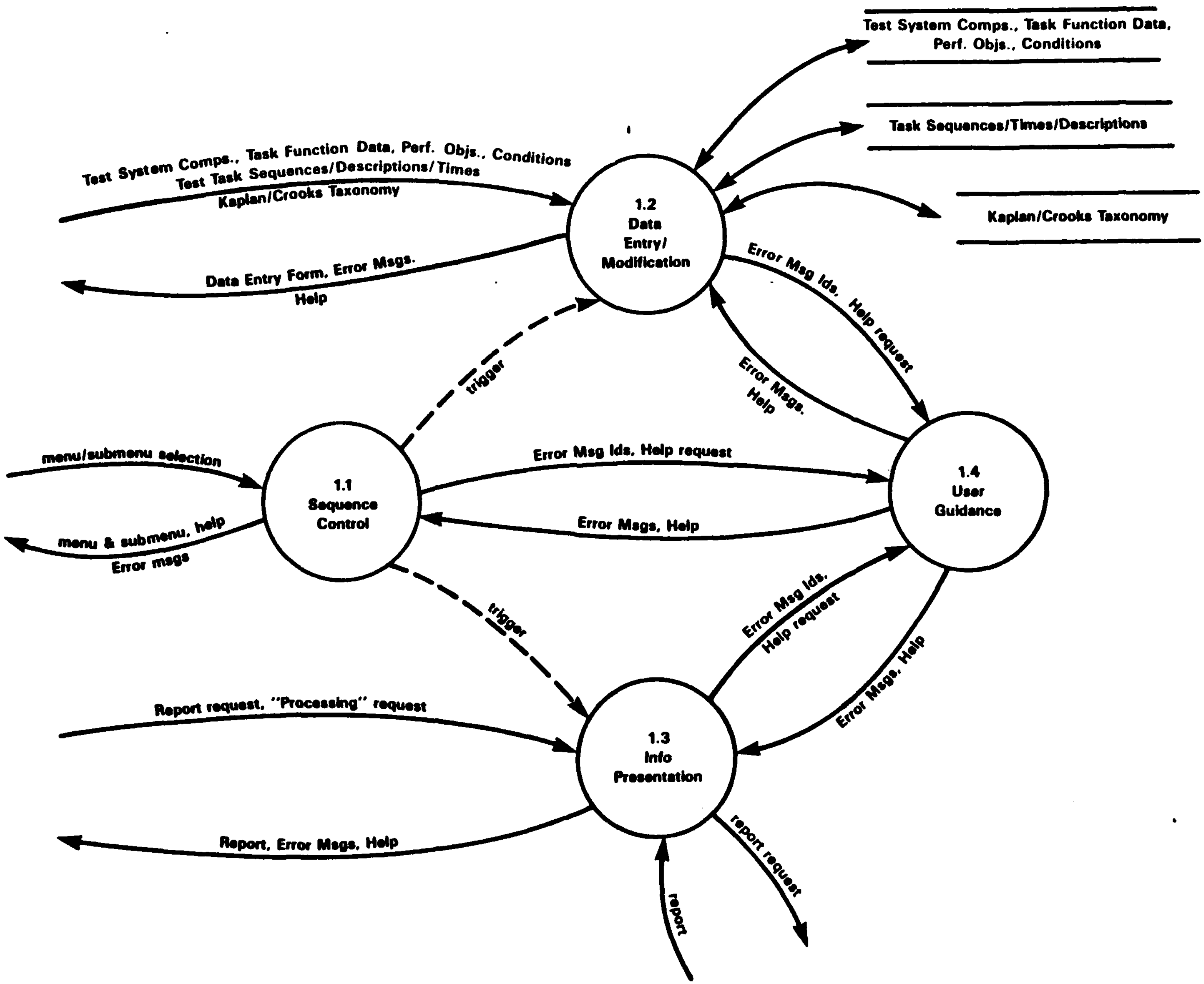


Figure 15. Level 2 Data Flow Diagram for User Dialog.

Guidance in the form of error messages and help requests related to data entry or modification.

The Information Presentation process interacts with User Guidance concerning error and help messages. It also interacts with the user in the report request sequence.

Derive Unique Jobs. Figure 16 presents the Level 2 data flow diagram for Derive Unique Jobs. The processes involved are: Classify Tasks; Establish Precedence Relationships; Test Feasibility; and Assign Tasks to Jobs.

Classify Tasks will group the tasks according to the way time is used to specify required performance. Category 1 tasks are those operator tasks with performance objectives related to response time requirements (e.g., time on target). Category 2 tasks are those maintenance tasks with performance objectives related to maintaining a constant rate or frequency of activity over some designated time period. Tasks with performance objectives related to maintaining constant activity over some designated time period (e.g., supervising monitoring, guarding) will be considered as "add-ons" to the operator or maintainer jobs most closely related. This categorization is necessary because the way in which jobs are formed differs depending on the type of performance objectives to be addressed.

Establish Precedence Relationships involves organizing and coding the tasks to reflect the sequence in which tasks must be completed in order to properly achieve the performance objective. This relationship is necessary for Category 1 tasks only. This process will be accomplished by developing a precedence network that shows which tasks must be completed before a given task can begin.

Test Feasibility determines the "critical path" through the network of tasks in order to determine whether or not the performance objective can be achieved given the task sequence and task times. If the critical path time exceeds either the response time required (for operator tasks), the user is informed that the performance objective can not be achieved and is transferred out of the job forming process so that either task times or sequence can be revised or the performance objective can be relaxed.

There are two types of tasks. Category 1 tasks are time-based, mission-oriented operator/field personnel tasks. These tasks must be completed within a specified time. Category 2 tasks are output-based, maintainer tasks (e.g., inspect, remove) and can be aggregated into maintenance ratios that are compared to the maintenance performance criteria (also in maintenance ratios). These tasks are performed continuously over time and result in the production of some countable output (e.g., parts replaced). A third task type, not covered in the current Product 1 taxonomy but nonetheless important are cognitive or monitoring tasks. These tasks are performed constantly, but do not result in measured output and include tasks such as surveillance, security, and supervision. These may be operator or maintenance tasks.

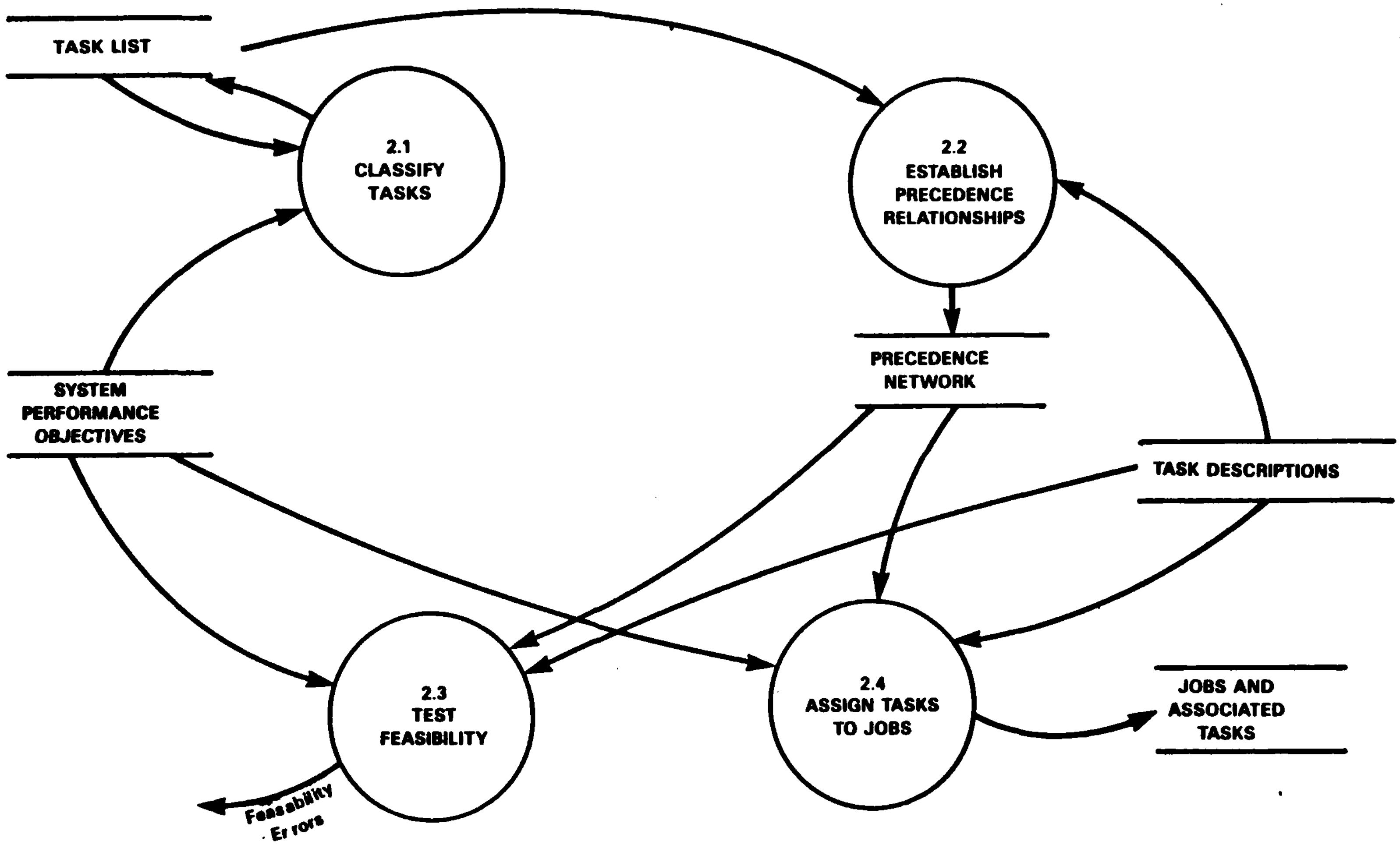


Figure 16. Level 2 Data Flow Diagram for Derive Unique Jobs.

Job construction using Category 1 tasks will be accomplished using Brook's algorithm. (A narrative description for this process excerpted from the concept paper for Phase 1 of this project as well as a listing of FORTRAN code for the algorithm is presented in Appendix B. We will extract those portions of this code that support Product 5, and translate them to the "C" language of Product 5.) This process assumes 1) that tasks are ordered according to the amount of time each controls in the precedence network (i.e., the "critical path" time beginning with each activity), and 2) that tasks are assigned to jobs such that the required response time or production rate is achieved.

Job construction using Category 2 tasks will be accomplished by multiplying maintenance task times by their expected frequencies to determine total time (over a specified time period) required for each maintenance task. Maintenance tasks at each maintenance level will be summed to determine total maintenance manpower requirements.

Tasks such as management/surveillance or other cognitive tasks are overlayed on the jobs resulting from Category 1 and 2 tasks such that, to the extent possible, they are combined with jobs that already exist.

We felt that it was important to further define the "Assign Tasks to Jobs" process in the Level 2 data flow diagram. Figure 17 presents the Level 3 data flow diagram for this process. The Level 3 processes are: Determine Critical Path Times; Rank Tasks by Critical Path Times; Assign Tasks to Jobs Based on Criticality and Resource Availability (Category 1 jobs); Determine Total Time for Each Task at Each Maintenance Level, Compute Number of Maintainer Jobs at Each Level (Category 2); and Determine Potential for Combining Coverage Tasks with Existing Jobs. The three data stores, System Performance Objectives, Precedence Network, and Task Times, all input to the formation of operator and maintainer jobs.

Generate/Print Reports. Much of the functionality of Generate Reports is provided by R:BASE. Figure 18 presents the Level 2 data flow diagram for generate/print reports. The diagram includes one user-related process, select report type, and eight report-type processes. These report-type processes are: operator functions, tasks, and times; operator task sequences; operator functions data; operator jobs and tasks; maintenance criteria; maintainer/subsystem/component data; maintainer component/task data, and maintainer jobs and tasks.

Structure Chart

Figure 19 presents the structure chart for the algorithm used for forming unique jobs. The inputs to the algorithm are task sequence, task times, and resource constraints. The algorithm calculates the critical path, that is, the path that traverses the network in the longest amount of time. The path incorporates user-entered constraints about simultaneity and single/multiple operator requirements. Next the algorithm assigns tasks to jobs, using tasks within a function, then taking tasks from the next most proximal function. The output is unique jobs and tasks with their times.

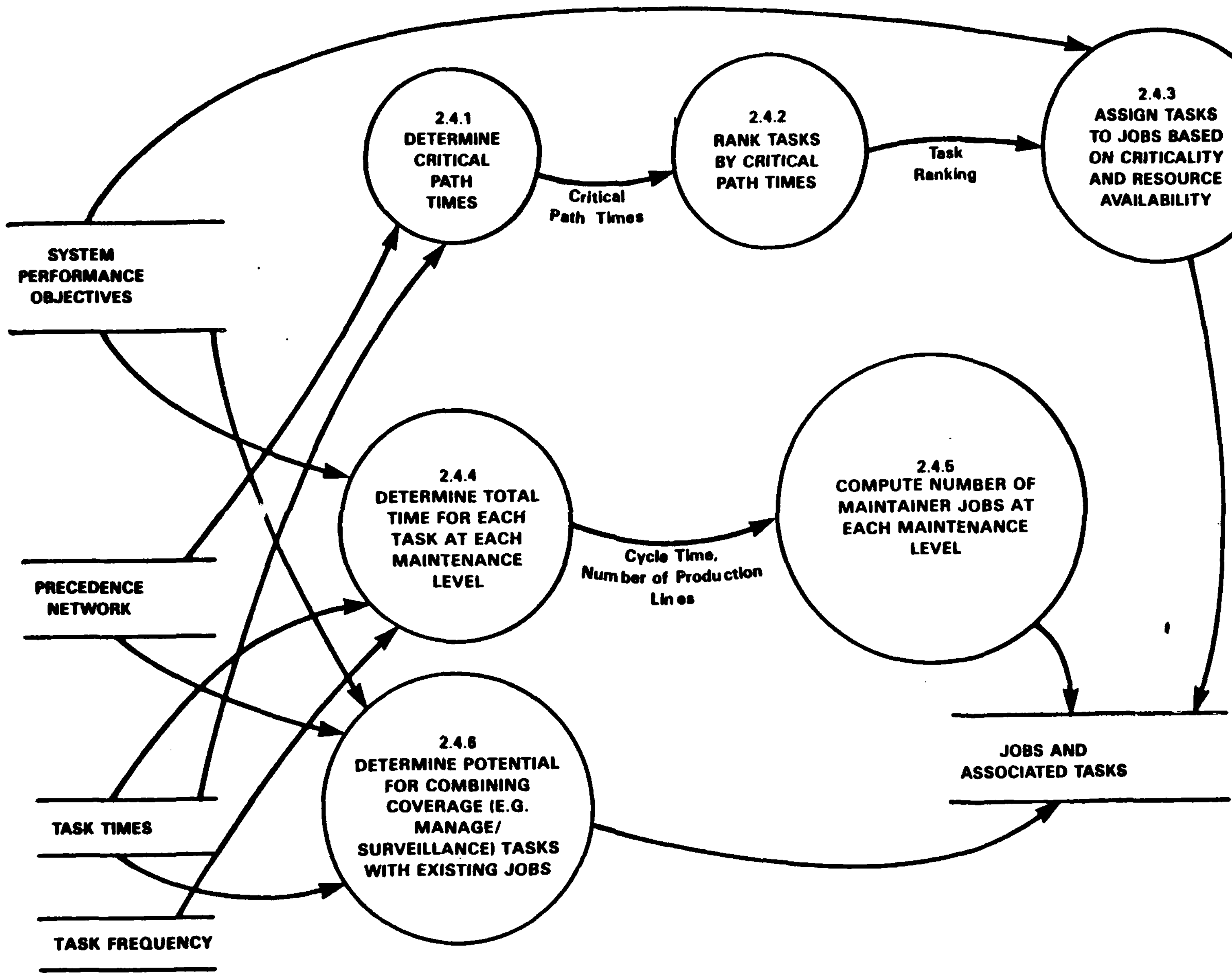


Figure 17. Level 3 Data Flow Diagram for Assign Tasks to Jobs.

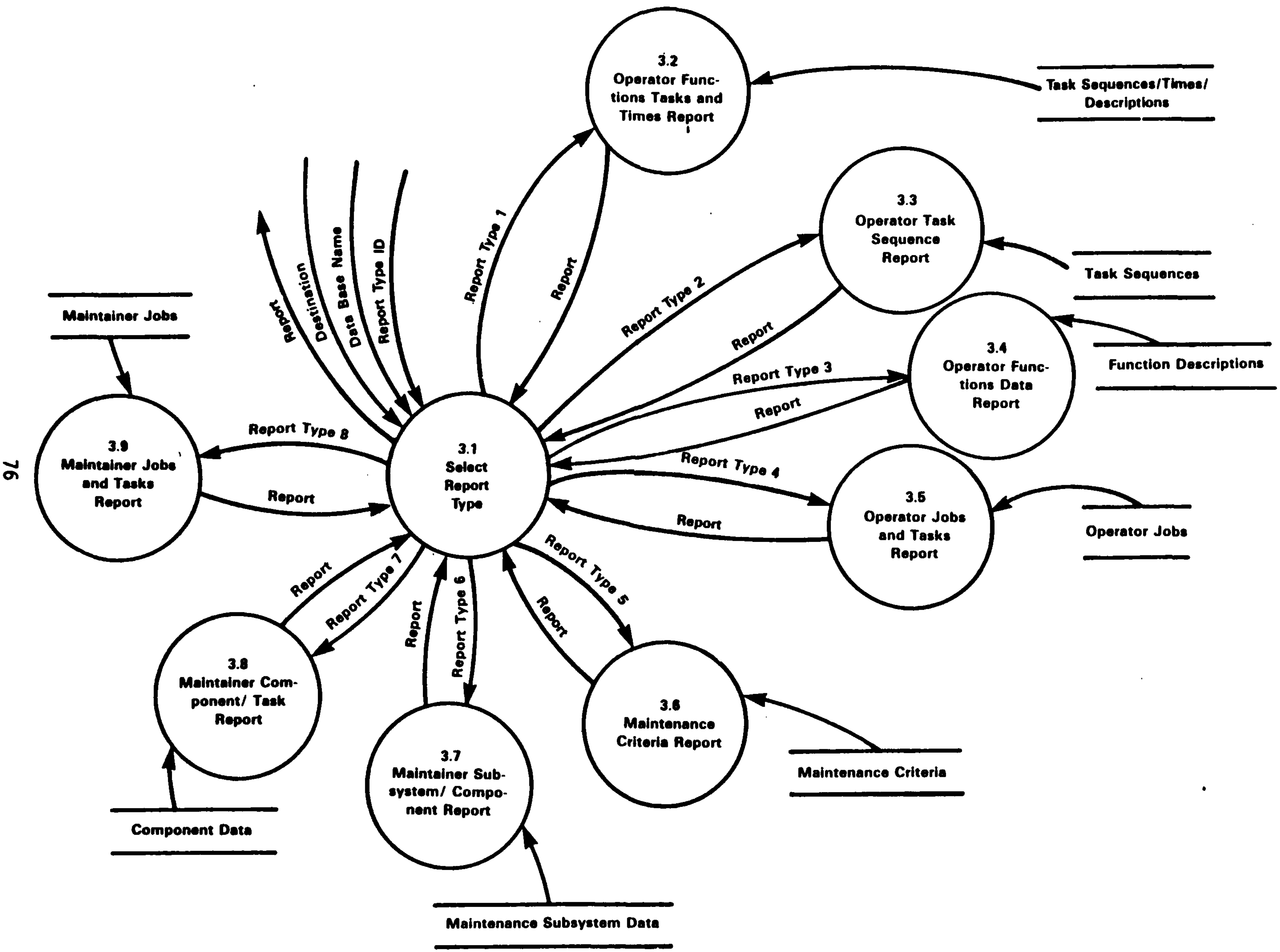


Figure 18. Level 2 Data Flow Diagram for Generate Reports.

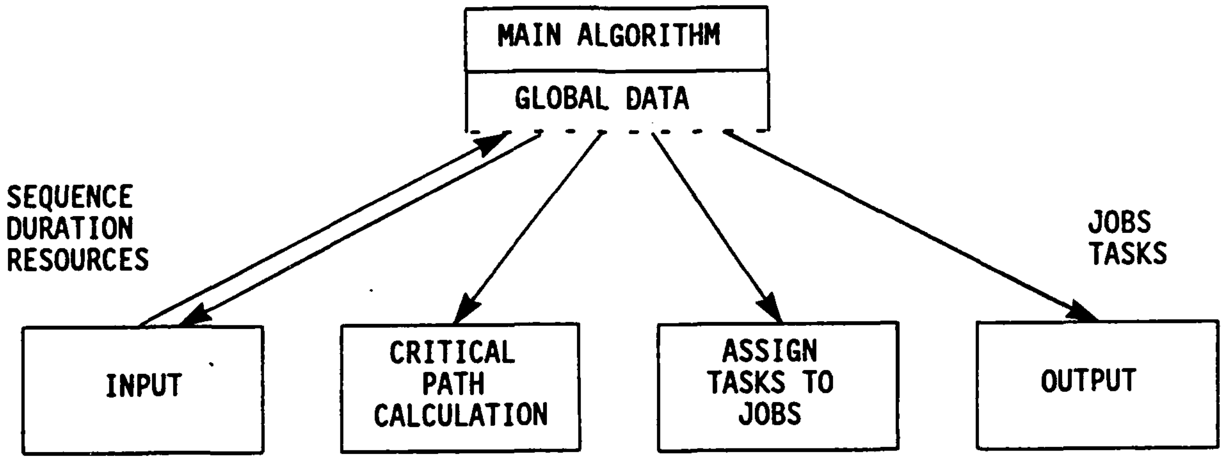


Figure 19. Structure Chart for Forming Unique Jobs.

We have considered two standard industrial engineering algorithms for the operator manpower calculation for Product 5. They are the Resource Allocation (RESALL) and Branch and Bound Assembly Line Balancing (BABALB) algorithms. We have decided to use the RESALL algorithm based on the following.

The RESALL program in the "Balance" mode determines the minimum number of jobs necessary to complete a category 1 (operator) function within a given response time. RESALL in the "Allocate" mode determines the minimum amount of time in which a given number of resources (of various types - up to 20) can accomplish a function. In both cases, RESALL assigns specific tasks to resource units (jobs), but the model as currently constructed does not track the tasks assigned to each resource unit. The BABALB program determines the number of workstations necessary to accomplish a function given a desired cycle time. However, this program assumes that cycles can overlap such that each workstation may be working on a different cycle of the function. Consequently, the assignment of tasks to workstations given by BABALB is appropriate for functions with "production" requirements (e.g., maintenance tasks), but not those with "response time" requirements. The task assignments for response time (category 1, operator) tasks will have to accomplished by modifying the RESALL program to compare the actual task assignments to resource units. This approach will give a feasible solution to the problem.

Integration of R:BASE System V

Product 5 is primarily an information-based application that requires a robust user-interface to ease use by analysts. As such, many of Product 5's requirements can be achieved readily through the utilization of a commercial off-the-shelf data base management system. Dr. Kaplan of ARI has encouraged the contractor teams to use the same data base management system to promote consistent user interfaces among products. We have elected to use R:BASE System V by Microrim, a data base management system chosen by other contractor teams.

Many of the significant decisions regarding the approaches for developing Product 5, as well as design implementation decisions for the operational Product 5, are directly related to the integration of R:BASE. A proper software solution for Product 5 (as well as other products) will integrate R:BASE application development and operational capabilities. The following discussion overviews those capabilities of R:BASE which will be integrated into the developmental and operational aspects of Product 5.

Application Development. R:BASE provides application development tools to define menus/submenus, as well as forms for data base entry/modification and report generation. These are implemented in separate programs that interactively guide the developer through definition dialogues, after which R:BASE procedural language code may be generated. Subsequent modifications to the generated code can be made either automatically (using the interactive definition dialogue), or manually (to customize).

The application development tools of R:BASE will aid the development of Product 5 considerably. They will permit the rapid development of prototype versions of Product 5 (with increasing functionality). This prototyping will enable ARI to become more involved in Product 5 development by providing recurrent feedback to developers as the implementation evolves.

Application Express is R:BASE's tool for creating menus, organizing them into a tree of menus/submenus, and for associating actions (other than submenus) to menu options. These other actions include: entering, modifying, and displaying data using a form; printing a report; displaying a help screen; and invoking an R:BASE procedure or external language (e.g., "C") program.

Menu options can be defined to be displayed both horizontally and vertically. Users make menu selections by moving the cursor direction keys or striking the number corresponding to the option (horizontal options only), following by a carriage return.

Through Application Express, users also define data base records and their fields, as well as the types and precisions of fields.

Forms Express is R:BASE's tool for interactively defining forms used for data entry, deletion, or modification. Developers use a variety of function keys that correspond to actions which enable forms to be "painted."

Permissible options (e.g., add, modify) are associated with the form during form definition. The interactive dialogue prompts the developer to stipulate record attributes to be displayed, how they are to be sorted prior to being displayed, and conditions (attribute values) for selecting attribute values to be displayed. The conditions may be either hard-coded with the form or user-specified (at run time).

R:BASE's tool through which developers interactively define reports is Reports Express. As with the other application development tools, developers use a variety of function keys that correspond to actions which enable (here) reports to be defined easily. R:BASE reports are comprised of a number of reports sections (that are individually defined): the actual data to be extracted from the data base; report/page/break headers; and report/page/break footers. This variety of report sections enable complex and attractive reports to be interactively defined. Again, as with the other application development tools, the generated code can be manually customized.

Operational Capabilities. Through Forms Express, R:BASE provides a variety of mechanisms that will help to insure the integrity of data provided for entry/update to the underlying Product 5 data bases. These include testing 1) numeric data to be within a specified range, 2) character data to be of specified enumerated values, and 3) referential integrity against values in other tables. Other related R:BASE features will be used that define default values for fields and fields for which data must be filled, as well as double entry verification for data that are entered or modified. In addition, as data are displayed through a form,

users may move through instances of the single record type or extractions from multiple records types (views) using function keys.

Forms are displayed with menu name at top, so that users maintain a sense of "where they are." A status line is available at the bottom of a form screen for displaying messages relating to the success or failure of user-submitted operations.

R:BASE supports a rich set of 89 commands as part of its procedural command language. Most significant of these is the variety of commands used to navigate through and manipulate data in data bases. Further, R:BASE provides a set of 70 math and string functions that are available in the command language. Errors resulting from command or function executions (on behalf of users) can be trapped and acted upon (e.g., security violations logged).

Data Base Security

For data base security, R:BASE supports the notion of a data base owner (i.e., superuser), with ability to assign read and modify passwords to individual tables or views. Backup and load utilities are available for logging data base files and reconstituting versions of the data bases.

In its newest release R:BASE provides a run-time, host language interface from the C programming language. This set of routines will be used in those portions of Product 5 that do not lend themselves to being written in the R:base procedural command language. R:BASE also provides the "Filegateway" facility for importing/exporting data in a textual representation to/from the underlying data bases.

The security mechanisms provided by R:BASE: USER PASSWORD, OWNER PASSWORD, READ PASSWORD, and MODIFY PASSWORD will be used to implement Product 5 security. All users initiating use of Product 5 are prompted for a USER PASSWORD (see Figure 6), which makes that user known to R:BASE. System description data bases are created with OWNER PASSWORD equal to USER PASSWORD.

READ PASSWORD and MODIFY PASSWORD, rather, are explicitly established by system description data base definers (owners). These enable users to protect their data bases (in their workareas) from unauthorized access by other users.

When a user specifies his or her intentions to use an existing data base, a knowledge of whether he/she owns that data base is ascertained by R:BASE by comparing the USER PASSWORD to the OWNER PASSWORD of the target data base. Given the data base to be used is not owned by the user and the user proceeds to attempt to read the data base, Product 5 will prompt for the user to specify a READ PASSWORD to which the USER PASSWORD is then temporarily assigned. R:BASE will then not not allow a user to read portions of the data base unless the USER PASSWORD is equal to the READ PASSWORD established for the data base. The mechanism to protect other users from modifying a data base owned by a user is accomplished in a

analogous manner using the MODIFY PASSWORD. The utilization of READ/MODIFY enables owners to assign different READ/MODIFY PASSWORDs for each data base, as well the ability to modify existing READ/MODIFY PASSWORDs.

User Workareas

User workareas, run-time components of R:BASE System V, the Kaplan/Crooks Taxonomy, and elements of the Product 5 application (e.g., menus, forms) will be segregated to different directories in the hierarchical file system.

A "\users" directory will be established, beneath which a subdirectory will be established for each user using his/her name (i.e., for user 1: "\users\'user 1 name'"; for user 2: "\users\'user 2 name'", etc.). This provides a separate workarea for each user, beneath which further subdirectories are created to maintain data bases generated by separate Product 5 "runs" for a specific user. These subdirectories are assigned the name of the data base prompted for by Product 5 when a new run is initiated.

Within these subdirectories ("\users\'user name'\'data base name'\"), the actual R:BASE data base for a new run is maintained (1 System Type data base), as well as all reports generated from that data base. By default, these reports are stored in file names that identify the type of report content (e.g., "oftt.rpt" corresponds to the "Operator Functions, Tasks, and Times" report, although users may specify target files of their own choice.

R:BASE maintains information for a data base in three files: 1) the (actual) data base, 2) the data base structure, and 3) the indices. Only the actual data bases are maintained in separate user workareas. Files containing the data base structure and indices are shared by all users and maintained in the directory containing elements of the Product 5 application.

Data Base Entity Relationship Diagrams

Entity modeling (Teorey & Fry, 1982) has been used to formalize the analysis of Product 5 data. This methodology employs entity relationship diagrams and entity definitions.

Entity relationship diagrams are used to graphically express the relationship between data objects. In these diagrams, data objects in the data bases may take any of five relationships. These are one to one (1:1), one to many (1:M), many to one (M:1), many to many (M:M), or not related. Data objects not related are not connected with arrows. 1:1, 1:M, M:1, and M:M relationships are shown with arrows. A single arrowhead denotes the "1" side and a double arrowhead denotes the "M" side of relationships.

Product 5 data bases employ high-level entity relationship diagrams for the Kaplan Crooks taxonomy (or whatever taxonomy is chosen) and Working/ Derived Data. These two entity relationships, each one presented from conceptual and implementation views, are presented in Figures 20-23, respectively.

Entity Definitions: The Data Dictionary

The data dictionary is developed from the entity relationship diagrams, described above. The data dictionary is a listing in tabular form of all the data elements in the data bases with a definition of the attributes and properties (e.g., type, precision) of each.

Table 2 presents the data dictionary for Product 5. It contains four columns: RECORD/Field Name; Type/Precision; Range; and, Unit of Measure. The RECORD/Field name is an abbreviated identification. The Product 5 data base contains 32 records (names in all caps), each with associated fields of the records, 12 are associated with the taxonomy, the remainder with working/derived data. The asterisk indicates a record type key (sometimes composite keys), which uniquely identifies a record type instance. The "Type/Precision" indicates type I, F, or C, for integer, fixed, or character string. Precision of an integer indicates how many digits are required. Fixed are expressed as x:y, where x:y indicates digits to the right and left, respectively. Precision of a character string indicates the number of characters allowed. "Range" indicates allowed values. Product 5 "Units of Measure" are seconds, times per second, and percent.

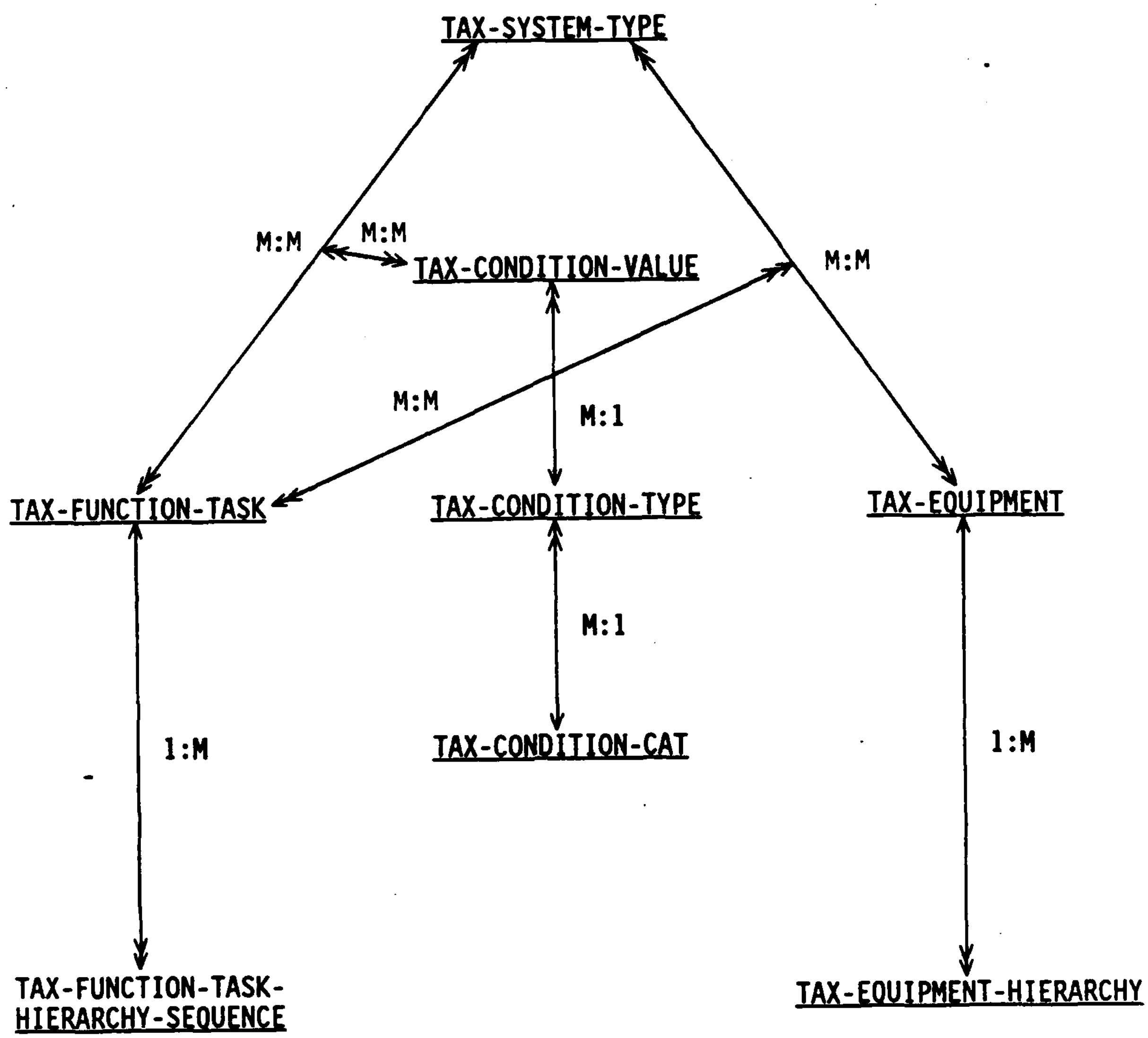


Figure 20. Kaplan-Crooks Taxonomy - Conceptual View.

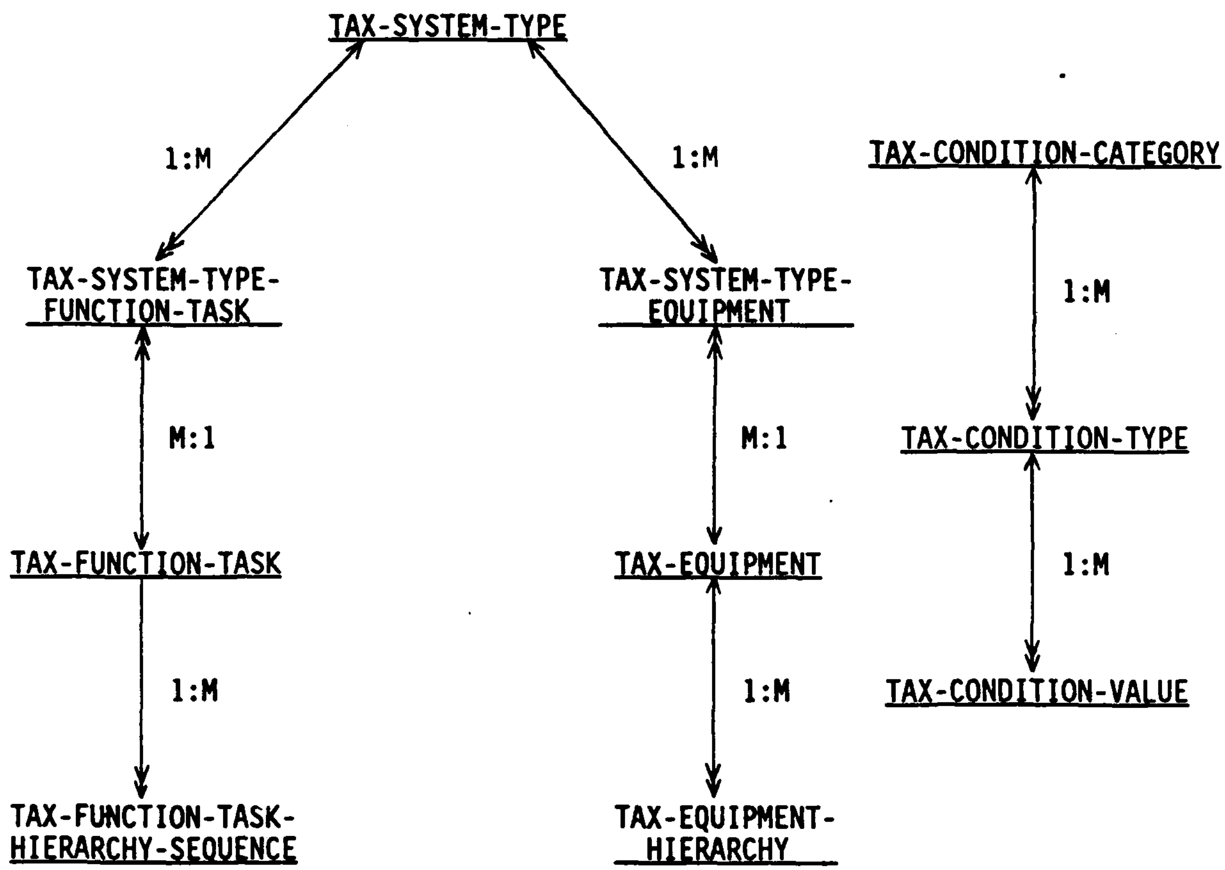


Figure 21. Kaplan-Crooks Taxonomy - Implementation View (1 of 2)

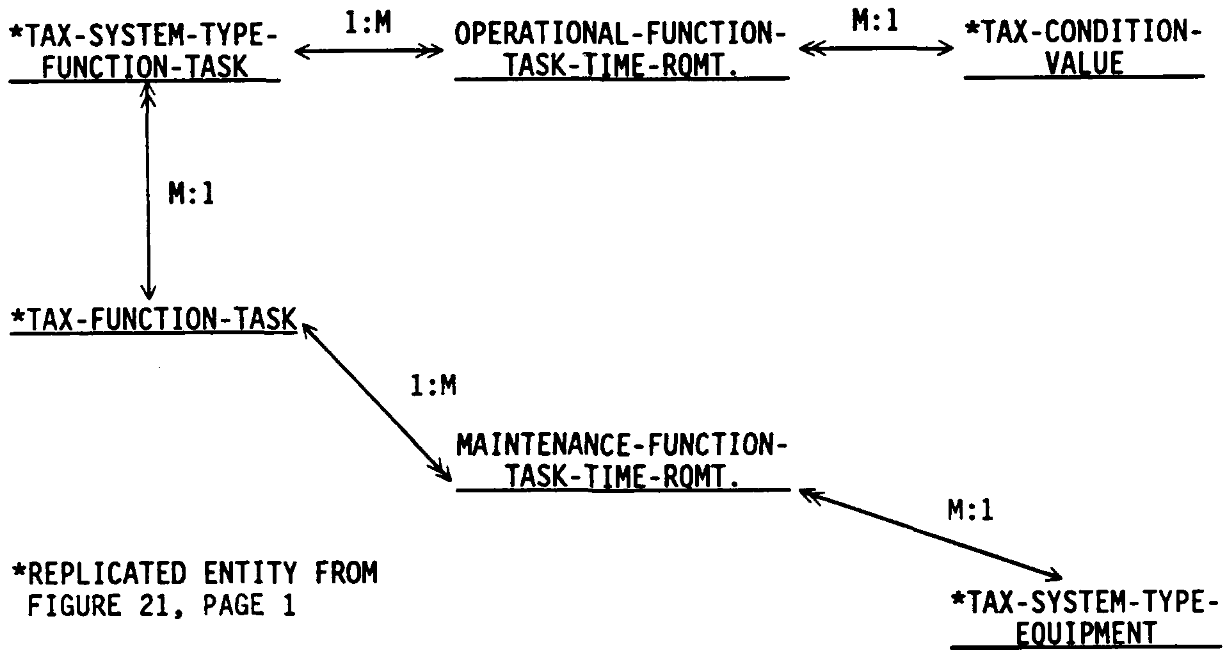


Figure 21 (Continued). Kaplan-Crooks Taxonomy - Implementation View (2 of 2)

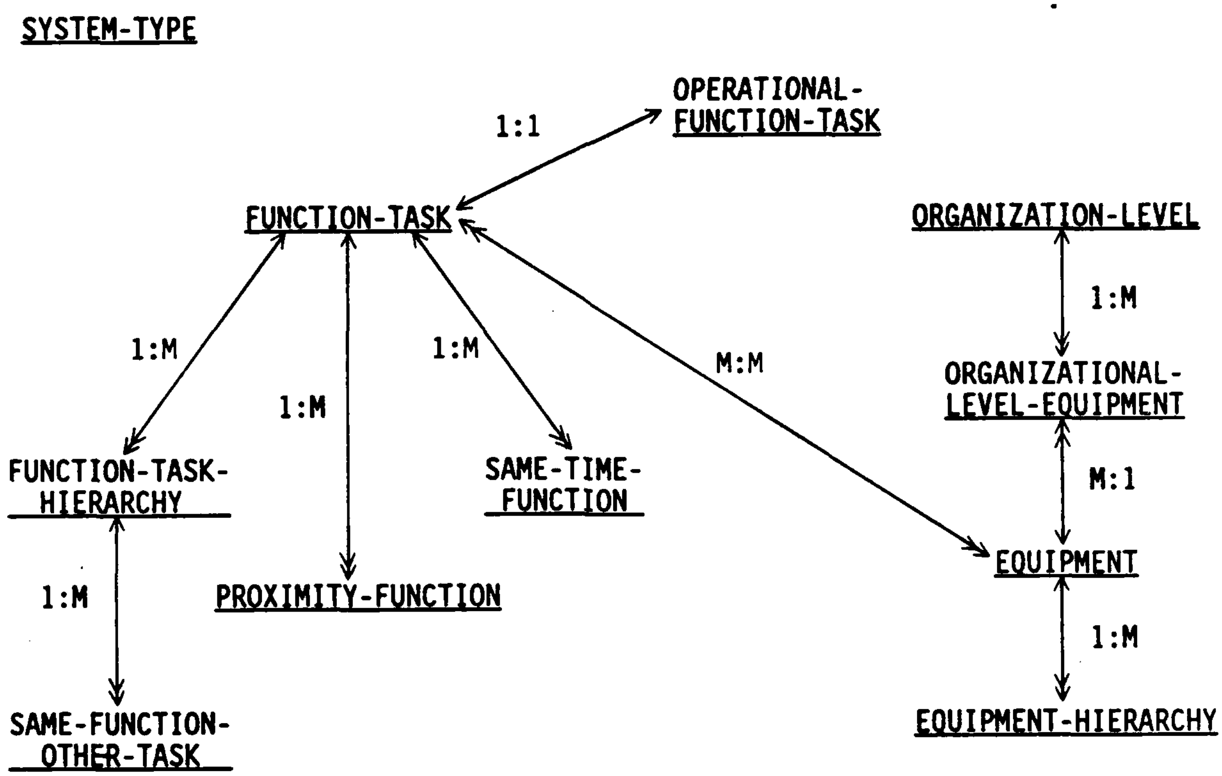


Figure 22. Working/Derived Data - Conceptual View (1 of 2).

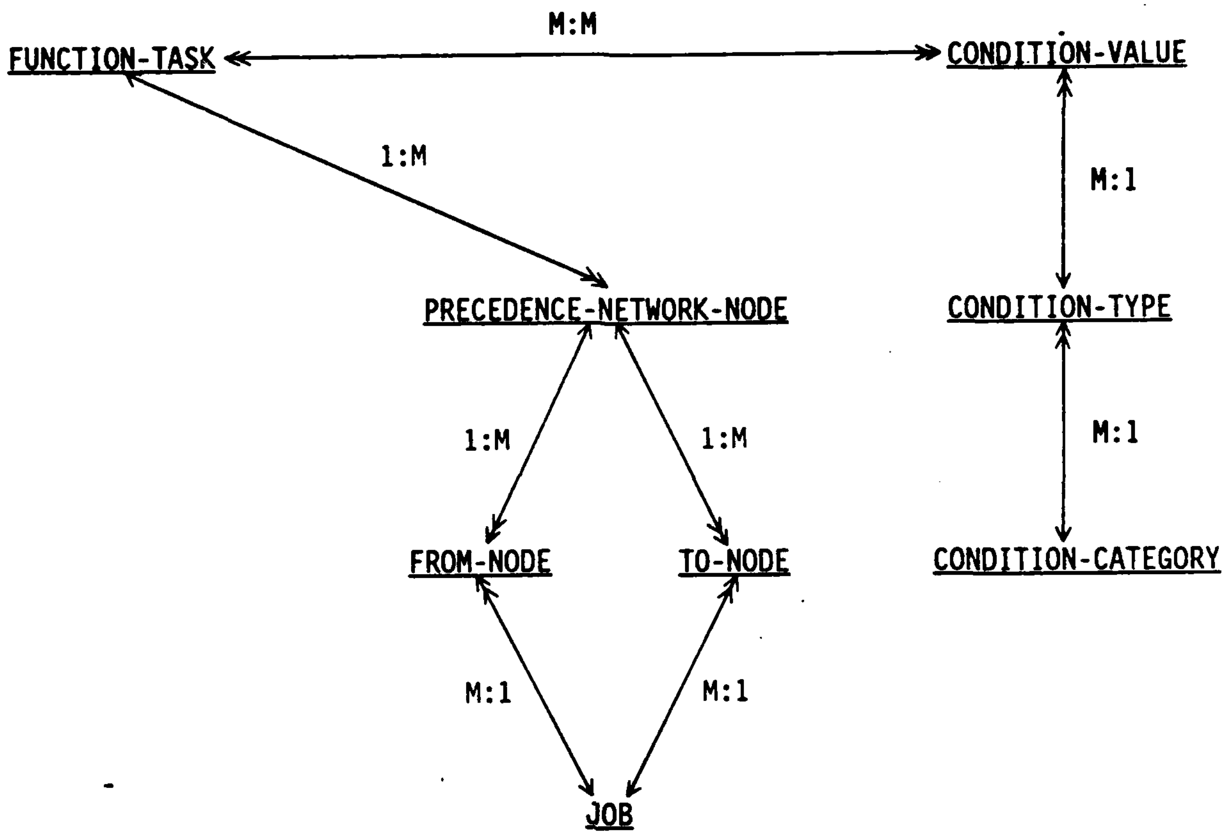


Figure 22 (Continued). Working/Derived Data - Conceptual View (2 of 2).

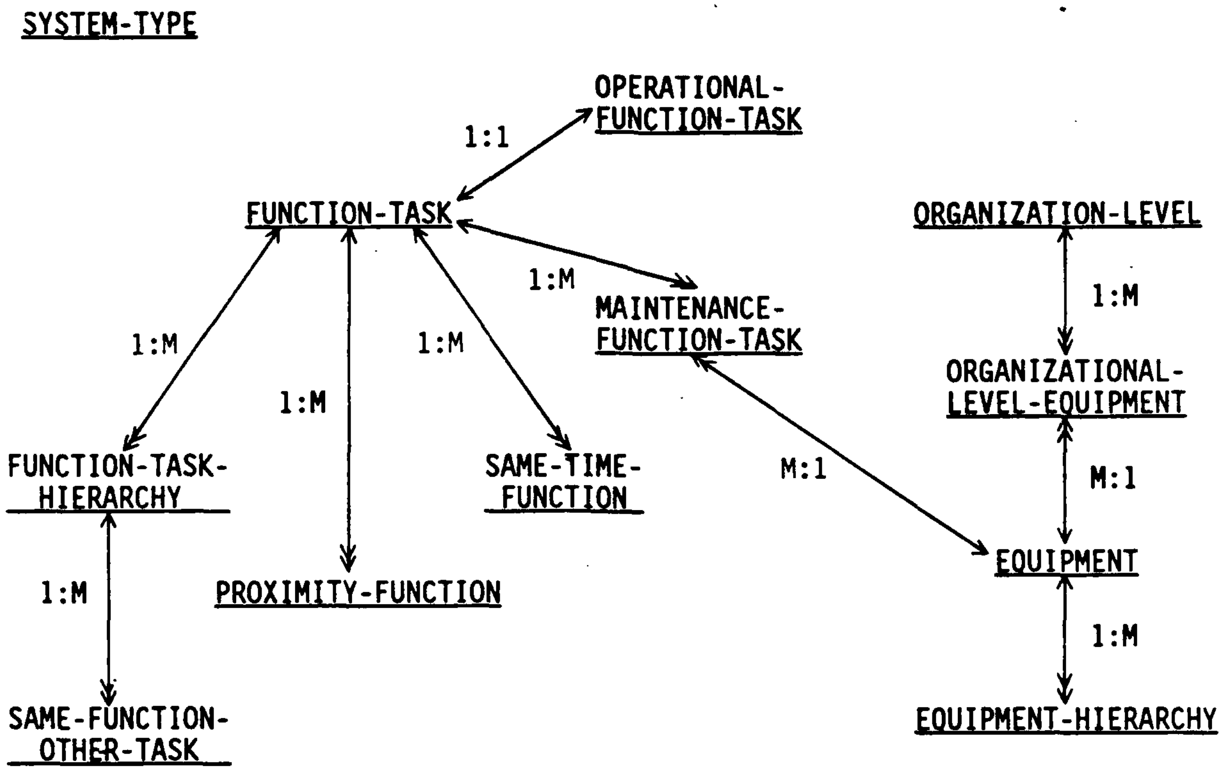


Figure 23. Working/Derived Data - Implementation View (1 of 2).

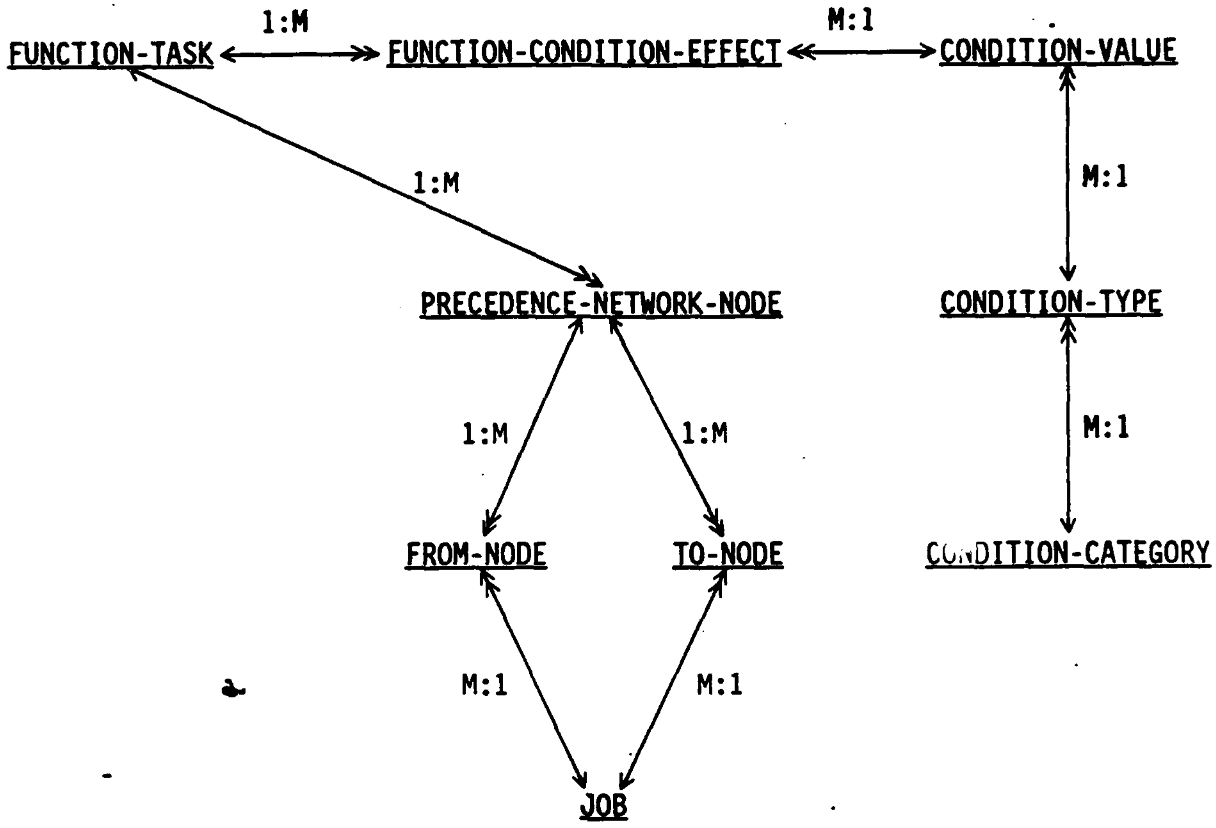


Figure 23 (Continued). Working/Derived Data - Implementation View (2 of 2).

Table 2. Product 5 Entity Definitions.

RECORD/Field Name	Type/Prec.	Range	U/M
TAX-SYSTEM_TYPE			
* System_type_id	I/2	1-21	n/a
System_type_name	C/30	n/a	n/a
TAX-SYSTEM_TYPE-FUNCTION_TASK			
* System_type_id	I/2	1-21	n/a
* Func_task_id	I/4		n/a
TAX-FUNCTION_TASK			
* Func_task_id	I/4		n/a
Func_task_name	C/30	n/a	n/a
Func_task_type	C/1	O,M	n/a
TAX-FUNCTION_TASK-HIERARCHY-SEQUENCE			
* Func_task_id	I/4		n/a
* Child_func_task_id	I/4		n/a
Task_sequence	I/2		n/a
TAX-SYSTEM_TYPE-EQUIPMENT			
* System_type_id	I/2	1-21	n/a
* Equipment_id	I/3		n/a
TAX-EQUIPMENT			
* Equipment_id	I/3		n/a
Equipment_nm	C/30	n/a	n/a
Criteria_maint_ratio	F/3.2		
TAX-EQUIPMENT-HIERARCHY			
* Equipment_id	I/3		n/a
* Component_equipment_id	I/3		n/a
TAX-CONDITION_CATEGORY			
* Condition_category	C/20	n/a	n/a
TAX-CONDITION_TYPE			
* Condition_type_id	I/3		n/a
Condition_category	C/20	n/a	n/a
Condition_type_name	C/30	n/a	n/a

Table 2 (Continued). Product 5 Entity Definitions.

RECORD/Field Name	Type/Prec.	Range	U/M
TAX-CONDITION_VALUE			
* Condition_value_id	I/3		n/a
Condition_type_id	I/3		n/a
Condition_value_name	C/30	n/a	n/a
OPERATIONAL_FUNCTION_TASK_TIME_RQMT			
* System_type_id	I/2	1-21	n/a
* Task_id	I/4		n/a
* Condition_value_id	I/3		n/a
Performance_reqmt_time	I/6		secs

-- Operational performance requirements are defined by a combination of System Type, Task, and Condition.

RECORD/Field Name	Type/Prec.	Range	U/M
MAINTENANCE_FUNCTION_TASK_TIME_RQMT			
* System_type_id	I/2	1-21	n/a
* Equipment_id	I/3		n/a
* Task_id	I/4		n/a
Performance_reqmt_time	I/6		secs

-- Maintenance performance requirements are defined by a combination of System Type, Equipment, and Task.

RECORD/Field Name	Type/Prec.	Range	U/M
SYSTEM_TYPE			
* System_type_id	I/2	1-21	n/a
System_type_name	C/30	n/a	n/a
Max_nm_operators	I/3		n/a
Max_nm_maintainers	I/3		n/a
FUNCTION_TASK			
* Func_task_id	I/4		n/a
Func_task_name	C/30	n/a	n/a
Func_task_type	C/1	n/a	n/a
Perc_crew_member_commtd	I/2	0-100	%
Number_operators_reqd	I/2		n/a
FUNCTION_TASK-HIERARCHY			
* Func_task_id	I/4		n/a
* Child_func_task_id	I/4		n/a

Table 2 (Continued). Product 5 Entity Definitions.

RECORD/Field Name	Type/Prec.	Range	U/M
SAME_FUNCTION-OTHER_TASK			
* Func_task_id	I/4		n/a
* Child_func_task_id	I/2		n/a
* Same_func_other_task_id	I/4		n/a
Completed_before	C/1	Y,N	n/a
Same_soldier	C/1	Y,N	n/a
Different_soldier	C/1	Y,N	n/a
PROXIMITY_FUNCTION			
* Func_id	I/4		n/a
* Prox_func_id	I/4		n/a
Closeness_ranking	I/2		n/a
SAME_TIME_FUNCTION			
* Func_id	I/4		n/a
* Same_time_func_id	I/4		n/a
OPERATIONAL_FUNCTION_TASK			
* Func_task_id	I/4		n/a
Performance_reqmt_time	I/6		secs
Task_time	I/6		secs
ORGANIZATION_LEVEL			
* Organization_level	C/30		n/a
ORGANIZATIONAL_LEVEL-EQUIPMENT			
* Organization_level	C/30		n/a
* Equipment_id	I/3		n/a
EQUIPMENT			
* Equipment_id	I/3		n/a
Equipment_nm	C/30	n/a	n/a
Quantity	I/3		n/a
Criteria_maint_ratio	F/3.2		
EQUIPMENT-HIERARCHY			
* Equipment_id	I/3		n/a
* Component_equipment_id	I/3		n/a

Table 2 (Continued). Product 5 Entity Definitions.

RECORD/Field Name	Type/Prec.	Range	U/M
MAINTENANCE_FUNCTION_TASK			
* Equipment_id	I/3		n/a
* Task_id	I/4		n/a
Performance_reqmt_time	I/6		secs
Task_time	I/6		secs
Frequency	I/3		n/a
Unit_of_measure	C/10	n/a	
CONDITION_CATEGORY			
* Condition_category	C/20	n/a	n/a
CONDITION_TYPE			
* Condition_type_id	I/3		n/a
Condition_category	C/20	n/a	n/a
Condition_type_name	C/30	n/a	n/a
CONDITION_VALUE			
* Condition_value_id	I/3		n/a
Condition_type_id	I/3		n/a
Condition_value_name	C/30	n/a	n/a
Default	C/1	Y,N	n/a
FUNCTION-CONDITION-EFFECT			
* Func_id	I/4		n/a
* Condition_value_id	I/3		n/a
Effected	C/1	Y,N	n/a
PRECEDENCE_NETWORK_NODE			
* Func_id	I/4		n/a
* Node	I/3		n/a
FROM_NODE			
* Func_id	I/4		n/a
* Node	I/3		n/a
* From_node	I/3		n/a
Task_time	I/6		secs
Job_id	I/3		n/a

Table 2 (Continued). Product 5 Entity Definitions.

RECORD/Field Name	Type/Prec.	Range	U/M
TO_NODE			
* Func_id	I/4		n/a
* Node	I/3		n/a
* From_node	I/3		n/a
Task_time	I/6		secs
Job_id	I/3		n/a
JOB			
* Job_id	I/3		n/a

USER ACCEPTANCE PLAN FOR PRODUCT 5

User Concerns

The purpose of a user acceptance plan is to identify potential sources of resistance to automation use and develop remedies for the problems prior to the introduction of the automation into the work setting. When a user is presented with an automated aid or tool for use in his or her job, that user is likely to ask a number of questions. The questions asked by a user will fall into two categories: (1) questions relating to getting started with the automation; and (2) questions relating to the performance with the automation. The questions relating to getting started with automation that a user might ask include:

- What is it going to take to get my paper files over to the computer?
- How much time is it going to take for me to learn to use this aid?
- How long will it be before I can actually get some work done with this aid?

These questions relate to the "start-up" costs associated with bringing new tools or techniques, particularly automated ones into the work environment. The areas of concern to the user relate to the transition and learning requirements associated with incorporating an automated aid into his or her work setting. In essence, the user is attempting to do a cost trade-off analysis, where "start-up" costs are typically paid for by time away from doing the day-to-day job. It is anticipated that the user's perceived "start-up" cost will be directly proportional to his or her resistance to incorporating the aid into the job.

Once a user is familiar with the use of the automated aid, other questions will arise relating to the performance of the aid, including:

- Is the aid doing something that I would rather do?
- What's going on inside the "black box"?
- Is the aid performing to my satisfaction?
- Is there enough time to accomplish my task with the aid?
- Is the aid improving the quality of my performance?
- Does this aid accommodate my increasing understanding and skill?
- How do my colleagues and supervisors view this aid?

Each of these questions reflects an area of user concern that may impact the acceptance or rejection of an automated aid in the job environment. These areas of concern are briefly discussed below.

Credibility. The first three questions relate to the credibility a user will assign to an automated aid. Credibility associated with automation may be partitioned into two aspects: belief that automation is capable of the functions allocated to it; and understanding in how automation is doing what it is doing. The two aspects of credibility are

associated with the allocation of functions to humans and automation and the user-computer interface facets of automation design, respectively.

During the front-end analysis phase of automation design, functions are allocated to automation for performance. If functions are allocated to automation that a user in the non-automated environment exerts control over, the credibility of automation performing such functions may influence the user acceptance of the automation. In particular, functions that require skill, judgment, or creativity may not be viewed by the user as credible if assigned to automation. This user perception may be based on his or her desire to retain control over the function or a perceived inadequacy of automation to perform the function. Regardless of the underlying reason, the result is a problem for the design developer in terms of gaining acceptance of the automation by the user.

The second aspect of credibility relates to the design of the user-computer interface. In particular, for automation to be credible to the user, the user needs some understanding of how automation is accomplishing the functions. Users cannot assign credibility of automation performance if the operation has the appearance of a "black box." The underlying issue here concerns the user's need to evaluate performance of an automated aid. The amount of information needed for evaluation is directly proportional to the expertise of the user, with the expert requiring the most and the novice requiring the least.

The ease of use of an automated aid is also likely to influence credibility assignment. If an automated aid is difficult to use the potential for user resistance of the aid is increasing. For example, an aid which forces the user to adopt new and different methods for accomplishing his or her task is likely to be resisted. Computer jargon is another example where the designer is likely to meet with resistance on the part of the user by asking that the user adjust to new or different terminology.

Quality of Job Performance. For an aid to gain user acceptance, the aid cannot be perceived as reducing the quality of the user's job performance. There are a number of ways in which the quality of job performance may be influenced by the introduction of an automated aid to the job environment. Functions allocated by the designer may result in user tasks that are viewed as unacceptable tasks to the user. In such a situation user rejection of the aid is likely. From a different perspective, unreliable performance by automation may require extra time and effort on the user's part to address the problems created; user acceptance is likely to be low for the perceived source of the problem, the automation.

Expertise Levels. There are two different types of expertise levels that should be considered in an attempt to gain user acceptance of an automated aid. First, there is the range of expertise inherent in the target user group. Second, there is the changing skill level of an individual user as he or she gains familiarity and proficiency with the use of the aid.

The concern about expertise level within the target user group focuses on the domain knowledge of users that the automated aid supports. When the

target user group is relatively homogeneous in their domain knowledge and skill levels, this concern is not a viable issue. On the other hand, if the target user group is heterogeneous, a potential problem exists. The underlying question facing an aid designer is "what level of proficiency in the job performance should we assume a user will have?" The higher the proficiency assumed, the fewer users in a heterogeneous group of users will actually be able to benefit from the use of the aid in their work environment.

The second part of the expertise level issue concerns the manner in which the user is able to interact with an automated aid as he or she gains experience. Novice users are only novices for a brief period of time. Frequently, friendly interface designs are optimized for the novice user and become a source of frustration as the user transitions from novice to experienced user. An automated aid which is designed solely for a novice user is only optimal for an environment where the aid does not become an integrated part of an individual's job environment. Such a situation would be characterized as a continually changing set of users who due to lack of repeated exposure do not transition beyond the novice stage of automated aid use. User acceptance of an automated aid requires that the transitional nature of user be accommodated unless the situation clearly indicates that users will not have repeated exposure or opportunities to use the aid.

Views of Colleagues and Supervisors. While the acceptance of an aid is not likely to made solely on the basis of the opinion of others, opinions will be an influencing factor. Enthusiasm among colleagues, in particular, for an automated aid in the job environment will help to lead to acceptance. Proponency for automation in the work place is frequently driven from the top-down. A top-down push for acceptance may work in the short term, but tends to be effective only while the proponent is in place; when the proponent leaves so does the enthusiasm.

The early involvement of the target users in the design and development process is one method for creating support for an automated aid. Users tend to view their involvement as an opportunity to achieve a positive influence on the impending changes to their work setting. The development of user interest groups serves as a mechanism for bringing the users into the process to benefit themselves as well as the automated design.

User Concerns and Product 5 Acceptance

To gain user acceptance of Product 5, it is important to identify the nature of the concerns discussed in the first section and their importance to Product 5 users. The first step is the identification of potential Product 5 users. When identifying the users of Product 5, we should consider that there may be two different sets of users. The most obvious set of users are those that will actually interact with Product 5 in the process of estimating manpower requirements for a target system. The concerns of this first set of users are likely to be those identified earlier in the section. A second set of users are those that will use the manpower estimates generated by the first user group. These individuals are

likely to be concerned with how the automation aids in arriving at the manpower estimate.

<u>First Look for and at Users</u>. Those users who are likely to directly interact with Product 5 in the process of estimating manpower requirements are likely to be found within the Directorate of Combat Developments at the numerous TRADOC schools. Preliminary efforts have been made to locate potential users at these locations. Specifically, contact has been established with individuals at Fort Rucker and Fort Knox based on names provided by TRADOC Headquarters. The purpose of exploring contacts at this stage of design is to begin to identify potential concerns for the user acceptance of Product 5. From our preliminary discussions with potential users a number of issues have surfaced that may influence the user acceptance of Product 5.

Individuals involved in manpower estimation are not likely to be prepared to devote an extensive amount of time transitioning to and learning to use an automated aid. One individual indicated that if he couldn't learn to use an aid in one day, he would be hesitant to use it. In terms of user acceptance, they are likely to balance the time required to learn to use the aid against the cost of time taken from their on-going job. In addition, at least some users will not be responsive to reading manuals, particularly lengthy ones, in order to learn to use the aid.

There is likely to be a heterogeneous population of user in respect to automation experience. Some users may be experienced with automation, though not necessarily from their job in the manpower estimation domain. Other users may have little or no automation experience. The design imperative of ease of use for Product 5 is on target for the potential users. There will be a need to accommodate the transitions of users as they become increasingly familiar with the use of the aid.

The process of manpower estimation is characterized as a time-consuming and difficult process. In some cases, users may have a large amount of data upon which to develop an estimate. In other cases, users may have sparse data available to them for developing an estimate. It is likely that users will be quite receptive to an aid that reduces the time consuming nature of their task.

Whether or not the target user group for Product 5 can be characterized as homogeneous or heterogeneous is not known at this point. However, it is apparent that there is no formal training for the manpower estimation process. There are publications from TRADOC, Soldier Support Center, as well as local expertise and possibly locally developed guidelines that provide the basis for on-the-job training. Attempts should be made early in Phase 3 of this effort to determine the range of domain knowledge represented in the potential user group of Product 5.

<u>Bring the User into the Design Process</u>. In an attempt to maximize the potential user acceptance of Product 5, users need to be brought into the design and development process. The beginning of Phase 3 of this effort is an optimum time to bring the user into the process. At such a time our design and development process will be a sufficient point of maturity to

provide users with the design of the aid for their comment. Importantly, bringing the user in at that point will still enable user modifications to become incorporated into the actual aid.

The suggested vehicle for bringing the users into the process is a user interest group. Currently, each school has a standing MANPRINT committee. While these committees do not necessarily contain those individuals who are likely to be the users of Product 5, the DCD members of such committees are likely in a position to identify the potential users. In addition, there are undoubtedly members on MANPRINT committees who would be interested to learn about Product 5 as potential users of Product 5 output.

The user interest group would have multiple objectives. One objective is to explain to users the purpose and proposed operation of the aid. The second objective is to elicit from potential users an evaluation of the proposed aid and suggested remedies to shortfalls in the design or improvements that may be made. The following topics should be addressed by design developers at a user interest group meeting:

- The objective and purpose of Product 5
- Potential benefits of using Product 5
- How Product 5 works including data entry requirements
- Product 5 interface operation
- How to judge the performance of Product 5
- How Product 5 adapts to various skill and experience levels

Next, data should be elicited from the potential users on the following issues:

- User concerns related to transitioning from their current method of manpower estimation to the use of Product 5.

- User concerns about the learning requirements associated with Product 5.

- Whether Product 5 will allow the users to perform the estimation process in a manner acceptable to them.

- The quality of job performance and whether or not Product 5 is viewed as an asset.

- Potential problems with the use of Product 5 as designed.

- Suggested improvements of Product 5.

There are a variety of ways that opinions could be elicited during a user interest group session. A questionnaire could be given to participants to insure that input is obtained from all interested members. The use of a questionnaire would be advantageous in that brief demographic questions could be included to differentiate comments of potential "hands-on" users from users of the Product 5 output. The questionnaire method for eliciting opinions has the advantage of developing a written record of concerns and the type of user raising the concerns.

Participants in the user interest group should be made aware of the impact they have on Product 5 design and development. Some sort of follow-up, such as a memo to the interest group, is advised to inform the users of actions taken on their concerns. In this way, bringing the users into the design and development process also means providing them with feedback to keep them in the loop.

Members of the user interest group should include users from each TRADOC school if possible. Accomplishing this objective might necessitate meeting with subsets of the group at different times and locations. The alternative of comprising an interest group from one to two schools could have the disadvantage of a lack of generalizability of the findings. While there are likely similarities in the manpower estimation process across the schools there may be some differences due to the type of systems or equipment of concern at each school.

Cost and Benefits of User Interest Groups. Bringing users into the design and development process has costs associated with it. As evidenced by the tentative agenda for a user interest group, advanced preparation of materials is necessary for the design developer team. Given the objective of eliciting input from all TRADOC schools, multiple user interest groups meeting would be a possible requirement. On the user side, there is a cost in the time for attendance. The actual duration of a user interest group meeting can be reduced by the preparation of "read ahead" packages. A "read ahead" package could contain: Product 5 description, possibly storyboards to show interaction, design developer concerns for which user opinion is needed, meeting objectives and agenda. The use of a "read ahead" package offers the attendees the opportunity to be prepared for the conference and frequently increases the quality of a conference while reducing the time required.

The benefits of a user interest group directly impact the potential user acceptance of Product 5. Allowing users to evaluate the design prior to implementation offers the benefit of identifying potential problems while the problems may be readily corrected. Membership in user interest group gives the user the opportunity to become part of the aid development team. By becoming part of the team the user has an investment in the ultimate successful implementation of the automated aid into his or her work environment.

REFERENCES

Bedworth, D.D. & Bailey, J.E. (1982). Integrated production control systems. New York: Wiley.

Brooks, G.H. (unpublished, 1963). "Algorithm for Activity Scheduling to Satisfy Constraints on Resource Availability."

De Marco, T. (1978). Structured analysis and system specification. New York: Yourdon Press.

Kaplan, J.D. & Crooks, W.H. (1980). A concept for developing human performance specifications. Aberdeen, MD: U.S. Army Human Engineering Labs, Tech. Mem. 7-80.

Page-Jones, M. (1980). The practical guide to structured systems design. New York: Yourdon Press.

Siegel, A., Pfeiffer, M., Kopstein, F., Wilson, L., & Ozkaptan, H. (1979). Human performance in continuous operations. Volume I. Human performance guidelines. Army Research Institute Technical Report. DTIC #AD086131.

Teorey, T.J. & Fry, J.P. (1982). Design of data base structures. Englewood Cliffs, NJ:Prentice-Hall, Inc.

APPENDIX A

SUMMARY OF R:BASE INTERFACE

Overview

The selection of the R:BASE data base management system (DBMS) for Product 5 directly constrains the nature of the Product 5 man-machine interface (MMI). This appendix identifies the specific MMI features of applications built using the R:BASE DBMS. These MMI features include the nature of menus and forms, as well as the keystrokes (and their sequences) used in manipulating menus and forms.

The Application Development Tools of R:BASE

R:BASE provides several application development tools through which users interactively define menus, associate objects to menu options, and define forms.

APPLICATION EXPRESS is used to define menus and to associate one of the following with each menu option (except "Exit"): 1) a form, 2) a lower level menu, 3) a "non-form-based" data base query/update statement in the R:BASE query language, 4) a report definition, or 5) an executable module written in a foreign (non R:BASE) language (e.g., "C"). APPLICATION EXPRESS is also used to define the structure of all data base tables which form the basis for the application.

FORMS EXPRESS is used to define forms, and is invoked by APPLICATION EXPRESS when the developer conveys the intention to associate a form with a menu option.

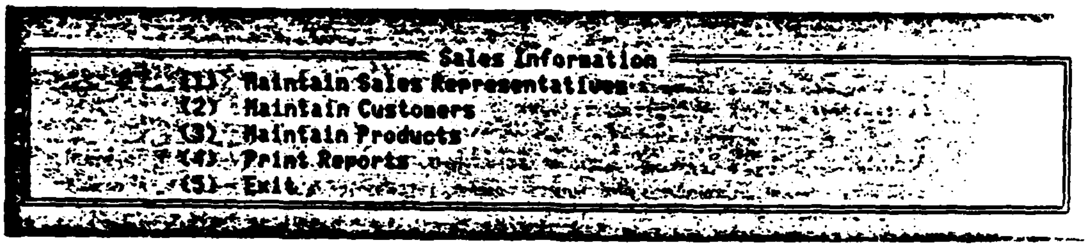
These are robust development tools. An additional tool, REPORTS EXPRESS is used to interactively define reports. All these tools generate "programs" in the R:BASE command language, which may be customized (e.g., through the insertion of additional statements in the R:BASE command language). A last step is required to compile the R:BASE command application definition into an executable representation.

Menus

R:BASE provides the mechanisms for defining two flavors of menus, vertically and horizontally-arranged menu options. Figure A-1 provides a sample of each. The last option in all menus should be "Exit." The selection of exit returns the user to the parent menu for all menu levels except the highest menu, and escapes the Product 5 application altogether at the main menu. R:BASE also enables the developer to optionally establish "[ESC]" with this menu "Exit" function.

For R:BASE menus with vertical options, the user selects an option by using the cursor direction keys (e.g., "down arrow") or typing the option

R:BASE Vertical Menu



R:BASE Horizontal Menu

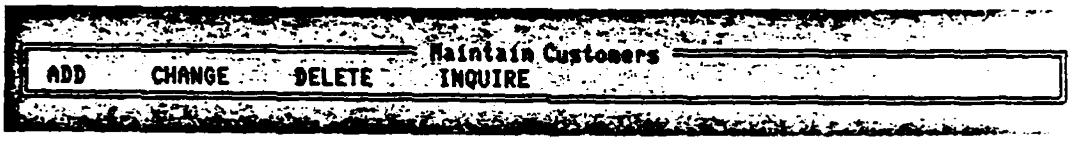


Figure A-1. R:BASE Vertical Menu and R:BASE Horizontal Menu.

number (immediate left of option), followed by a "[RETURN]." The developer can also define function keys to be associated with menu options.

Note that menus have a menu title that is displayed on the menu above options (in the sample vertically-arranged options menu, "Sales Information").

During menu definition, the developer can define help text to be associated with a menu, which is available to the user at run-time through the selection of "[F10]." For Product 5, this mechanism should be used to describe in a few sentences the purpose of each menu option. R:BASE enables this help text to be up to 5 pages in length, but Product 5 should constrain it to a single page.

R:BASE Forms

R:BASE forms have the following characteristics:

1. Form fields require explicit user entry to add/modify. The notion of highlighting items for selection is not supported.

2. Forms can map to at most 5 data base tables.

3. Components (see Figure A-2):

    a. Form title.

    b. Operations menu at top of form (R:BASE terms these "menu options"). Generally, users first complete form fields with data, and then select an operation from the top of the form (e.g., the equivalent of "insert," "update," etc.).

    c. Prompts for data entry (e.g., "Salesman:").

    d. Form fields (for data entry/display/modification) that are associated with either table fields or with variables.

    e. Status line which displays information about the completion status of data entry insertions/updates and field validation.

4. R:BASE supports the facility for expressing master-detail relationships on forms, such that fields from many instances of the detail record are displayed ("Transaction Detail" in Figure A-2) with a single instance of the master record to which they relate ("Transaction" in Figure A-2). This mechanism provides the remaining two components of R:BASE forms, below.

    a. "Tiers" which map to single instances of the detail record.

    b. "Regions" which are composed of the tiers and the detail record header line. There is a constraint of one region/form.

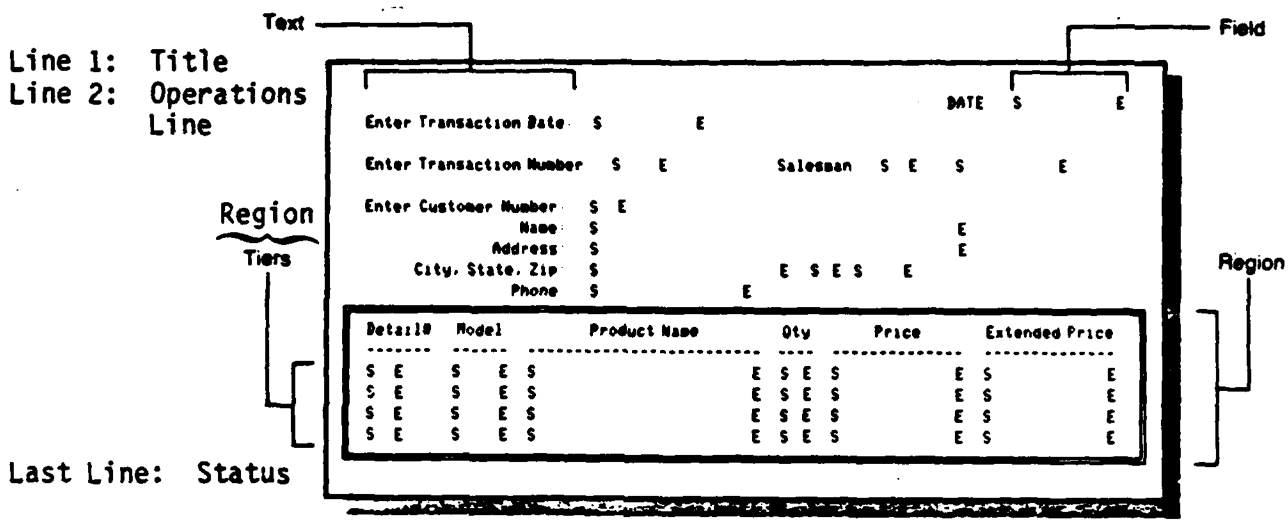


Figure A-2. Text, Fields, Regions, and Tier On a Form.

5. R:BASE forms are driven by either of two run-time components of R:BASE, ENTER and EDIT, which are associated with the form at form definition. ENTER is used when the primary purpose is to enter new rows of data (though it is also possible to edit rows using ENTER); EDIT is used when the primary purpose is to edit existing data (though it is also possible to enter new rows using EDIT).

6. There is a different set of menu operations associated with ENTER and EDIT (see Figure A-3 and A-4), that are displayed at the top of the form.

7. During form definition, FORMS EXPRESS prompts for 1) general form characteristics, 2) table characteristics, and 3) field characteristics. Although forms fields can map to as many as 5 tables, the first table specified is treated as the main table that the form is meant to serve. R:BASE defines default characteristics for fields based on the source of the field's value, so that form fields that map to data base fields in other than the main table are assumed for display only. These default field characteristics can be modified. Further, expressions that reflect table lookups to fill form fields in other than the main table must be defined explicitly when the form is used with either ENTER or EDIT; table lookups must also be defined for entries in the main table when the form is used with ENTER.

8. A number of function keys are defined that enable the user to move the cursor throughout a form and to otherwise manipulate data on a form. These function keys and their corresponding actions are identified in Figure A-5.

Product 5 Forms

R:BASE effectively differentiates between the user actions of initially entering data and then modifying it (after it has been written to the data base). As previously mentioned, there are different menu operations associated with both (Figures A-3 and A-4). Although ENTER can be used to modify data, ENTER operations do not include the capability to move backward and forward through instances of the main table the form is meant to serve (see "+" note at bottom of Figure A-5). Although EDIT can be used to enter new data, this is accomplihsed by "writing over" an existing instance of the target table on the form (followed by the "Add New" EDIT operation). The definition of "lookups" for a form also differ depending upon whether the form is used to enter or modify existing data.

Consequently, Product 5 should have different menu options and form definitions that correspond to the actions of initially creating new data, and subsequent modification of that data. Modification can also include the random insertion of new instances of some target table.

Table 7 Menu Options With the ENTER Command

Option	Purpose
Add	Adds the data on the form to the appropriate tables. Clears the screen for you to enter another row.
Duplicate	Adds the highlighted row as a new row to its table and leaves the values displayed in the fields for you to use again. If you have entered data on the form below the highlighted row, those rows are added to their tables and cleared from the screen. When you are entering repetitive information into a table, use this option to save time and keystrokes.
Edit Again	Returns you to the form so that you can edit your data. Does not add the data to the database. Applies only before the data is added to the database.
Discard	Removes the highlighted row from the screen. If the form serves more than one table and the row is not in the last table, a prompt asks if you want to discard only the highlighted row, or the highlighted row and any dependent rows further down the form.
Quit	Ends the session of form use. You can also leave the form by pressing [ESC].

For a list of function keys used in form processing, see table 6 under the EDIT command in this dictionary.

ENT is the shortest form of the command name.

Examples

```
ENTER tranform FROM b:\transact\trans.dat
```

Uses the form *tranform* to load data from the external file *trans.dat* residing on drive *b*: in directory *transact*.

```
ENTER tranform FOR 1 ROW
```

Displays the form *tranform* and allows the user to enter one row of data to the first table served by *tranform*. The user can enter as many rows of data in subsequent tables as are appropriate for the one row entered in the first table. This option is convenient in applications that require other actions to take place after loading each complete entry.

```
ENTER tranform AT 5
```

Displays the form *tranform* at screen row 5.

Figure A-3. The ENTER Option.

ENTER Using a Form

SYNTAX

```
ENTER ─┬───────┬─ formname ─┬─────────────────┬─┬──────────────┬─
       └ USING ┘            ├ FOR n ROWS ─────┤ └ AT scrnrow ──┘
                            └ FROM filespec ──┘
```

Related Commands EDIT USING, SET AUTOSKIP

See Also Chapter 3, *User's Manual*

Purpose The ENTER command is used to add data to tables using the specified form (see the FORMS command in this dictionary).

Options *USING*: This word is optional.

AT scrnrow...: Draws the form on a specific row of the screen other than the first.

FROM filespec: Indicates that the data is entered from an external ASCII file rather than from keyboard entry. It can only be used with a single-table form.

FOR *n* ROWS...: Limits the number of rows entered to the integer number represented by *n*.

Comments This command displays a previously created form on the screen to be used for data entry. For instructions on how to set up a form, see the FORMS command in this dictionary.

FileGateway is the recommended method for transferring fixed field ASCII files into R:BASE; however, you can transfer fixed field ASCII files into R:BASE using a form with the ENTER command and the *FROM filespec* option. If adding data from a fixed field ASCII data file with rows less than or equal to 80 characters, define a form that matches column entry locations to file locations. If the rows are greater than 80 characters, use FileGateway. To add data to a table via the keyboard, omit the file specification.

When a form is created, the form designer determines which database actions are allowed on the specified tables and which menu options are displayed during form use. Table 7 shows all the menu options available with the ENTER command using a form.

Figure A-3 (Continued). The ENTER Option.

EDIT Using a Form

SYNTAX

```
EDIT USING formname [AT scrnrow] [SORTED BY collist] ...
... [WHERE condlist]
```

Related Commands EDIT, ENTER, SET AUTOSKIP

See Also Chapter 3, *User's Manual*

Purpose The EDIT USING command is used to view, change or update data, or delete rows that are displayed in the specified form (see the FORMS command in this dictionary).

Options *AT scrnrow...*: Draws the form on a specific row of the screen other than the first.

SORTED BY...: Sorts the rows by the column(s) you specify in the column list.

WHERE...: Limits the rows to be edited. In a form that serves more than one table, the WHERE clause applies to the first table in the form.

Comments This command displays data on the screen using a previously created form (see the FORMS command in this dictionary for instructions on how to set up a form). When the form is created, the form designer determines which database actions are allowed on the specified tables and which menu options are displayed during form use. Table 5 shows all the menu options available with the EDIT command using a form.

Figure A-4. The EDIT Option.

Table 5 Menu Options With the EDIT Command

Option	Purpose
Edit	Moves you from the menu to the form so that the data displayed on the screen may be edited.
Save	Saves the changes that have been made on the displayed data, highlights the first table served by the form, and displays the data from the next row of that table. The rows that you changed are replaced in the table.
Add New	Makes a new copy of the highlighted row in its table and retains the original row without changes.
Delete	Deletes the highlighted row from its table and clears the row from the screen. Before this action takes place, a prompt asks you to confirm the command.
Reset	Resets the values in the highlighted row to their state before changes were made. Applies only to the highlighted row before changes have been saved to the database. After modifications have been stored, *Reset* can no longer be used to restore that row.
Previous	If changes have been made in the displayed data, asks for confirmation to save the changes to the database. Highlights the first table served by the form and displays the data from the previous row of that table.
Next	If changes have been made in the displayed data, asks for confirmation to save the changes to the database. Highlights the first table served by the form and displays the data from the next row of that table. The rows that you changed are replaced in the table.
Quit	Ends the session of form use. You can also leave the form by pressing [ESC].

Table 6 shows the function keys available when using a form.

Examples In each example, the form can be used to perform predefined database actions on the specified tables.

```
EDIT USING tranform SORTED BY custid
```

Displays the form *tranform* with the rows for the first specified table in customer id order.

```
EDIT USING tranform WHERE custid = 100
```

Displays the form *tranform* with only the rows for the first specified table in which the customer id number is equal to 100.

```
EDIT USING tranform AT 5 WHERE transid EXISTS
```

Displays the form *tranform* beginning at the fifth screen row, and displays all the rows from the first specified table that contain a value in the *transid* column.

Figure A-4 (Continued). The EDIT Option.

Table 6 Function Keys Used In Form Processing

Key	Purpose
[F2]	Erases the contents of the field from the screen.
[Shift-F2]	Starting with the cursor position, erases to the end of the field.
[F4]	Causes the last character typed to be repeated when you press the right or left arrow key. Press [F4] again to stop repeating the character.
[F5]	Resets the value of the current field to its original state (undoes edits).
[F7]†	Displays the previous row in the current table.
[F8]†	Displays the next row in the current table.
[F9]	Highlights the next table served by the form and moves you to the first field of that table.
[F10]	Displays help for the current field or page.
[Shift-F10]	Displays more function keys.
[Ins]	Inserts a space at the cursor.
[Del]	Deletes the character at the cursor.
[↑]	Moves to the previous line in a field with more than one line.
[↓]	Moves to the next line in a field with more than one line.
[Tab]	Moves to the next field in the current row. From the last field in a row, moves to the first field.
[Shift-Tab]	Move to the previous field in the current row. From the first field, go to the last field.
[ENTER]	Within a row, moves to the next field. From the last field in a row, moves to the next table. From the last field in a region that displays more than one row of data at a time, scrolls to the first field in the next row.
[PgUp]	Moves to the previous page in a multi-page form.
[PgDn]	Moves to the next page in a multi-page form.
[ESC]	From anywhere on the form, returns you to the menu. From the menu, returns you to the system from which you entered the form—the R> prompt or your application.

† When the form is used with the ENTER command, these keys apply only to rows entered in a region that displays multiple rows or rows displayed through master lookups.

EDI is the shortest form of the command name.

Figure A-5. Function Keys Used in Form Processing.

APPENDIX B

FORMING JOBS FROM CATEGORY 1 (OPERATOR) TASKS

The process for forming jobs from Category 1 tasks is drawn largely from the production scheduling and resource planning literature. Figure B-1 shows the process of creating these jobs. The steps are listed below.

1. Determine the technological sequence of tasks required to perform each function. (User may update original entered sequence.)

2. Develop a precedence network defining the task relationships to the required functions.

3. Determine the time required to perform each task under each set of environmental conditions. (Input earlier, user may update.)

4. Identify required response time for the job function and check task times against response time requirement. If one or more task times exceed the response time, the task(s) must be redesigned or the response time must be relaxed.

5. Identify constraints on assigning tasks to jobs due to proximity and simultaneity requirements.

6. Using automated resource allocation techniques, create work stations/jobs based on the precedence network and response time requirements.

7. Test the sensitivity of the number of work stations/jobs to the response time requirement.

8. List the possible job assignments and resulting response times.

Input to Step 1 is the task sequence entered by the user during data entry. Each task related to a given system function is selected by the user along with its immediate predecessor(s) (i.e., the task(s) that must be completed before it can begin). Note that all tasks related to a given function will fall in the same task group. The system design will drive the task sequence. The sequence will be determined by successively asking the question "What tasks must be completed before this task can begin?" The questioning process continues until all tasks related to a function have been placed in sequence (note that some tasks or series of tasks may be performed in parallel).

Step 2 formalizes the information collected in the first step by creating a network that reflects the aggregate set of precedence requirements associated with the successful accomplishment of a given function. The precedence network is important in that it identifies those tasks that must be performed in sequence and those that can (but not must) be done at the same time. This is done automatically.

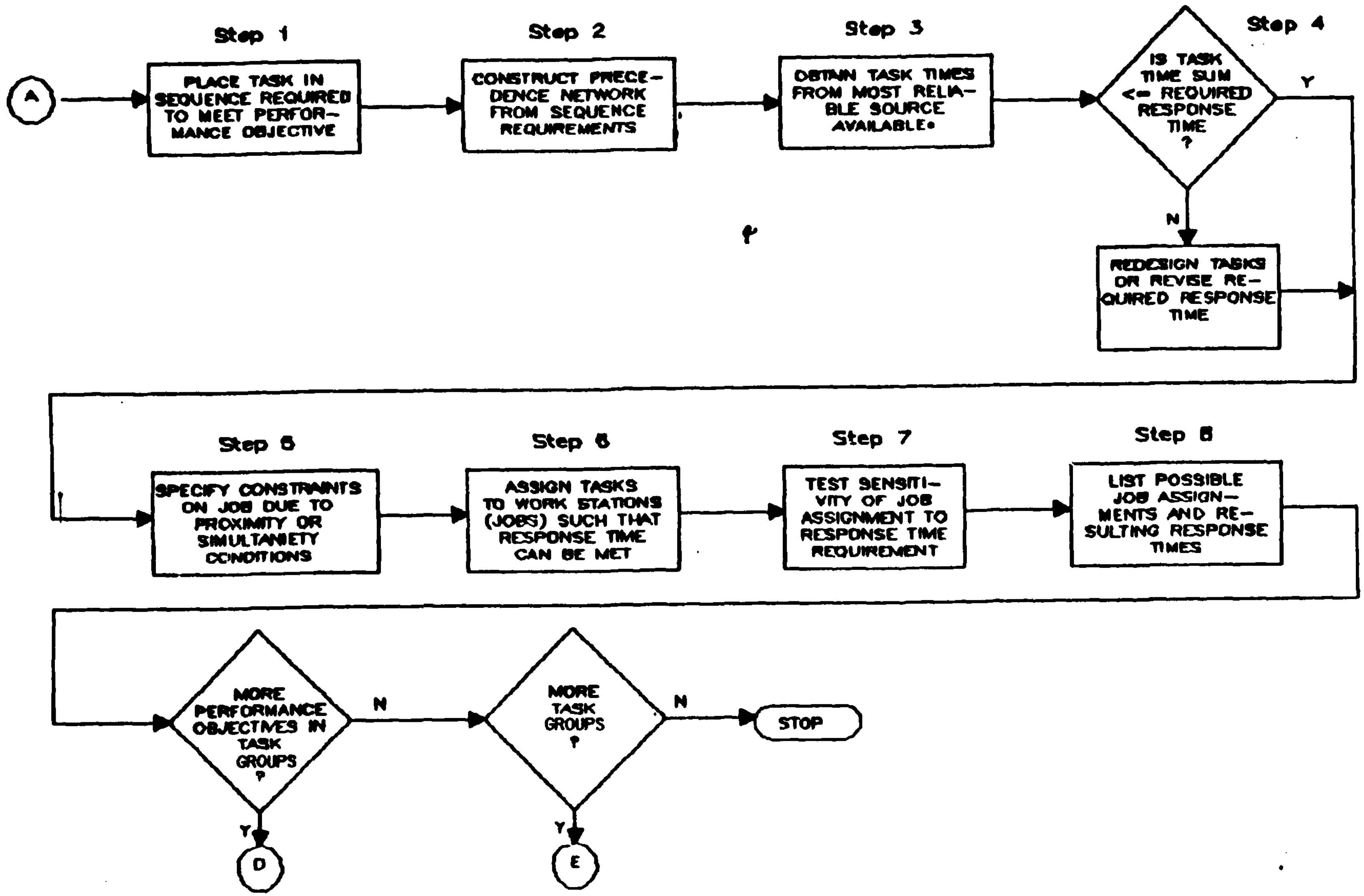


Figure B-1. Process for Converting Category 1 Tasks Into Jobs.

Step 3 assigns times to each of the tasks in the precedence network. The method that Product 5 will use to identify and assign task times was discussed earlier. An example of a precedence network (from Step 2) with task times (from Step 3) is shown in Figure B-2.

In Step 4, the response time requirement for the total job function is identified for the specific set of tasks required to accomplish the function. Note that the achievable response time for a function cannot be less than the greatest sum of all sets of required tasks that must be performed sequentially. If the sum of the task times for a required set of sequential tasks exceeds the required response time, then the response time cannot be achieved regardless of the crew size. In this case, either the response time must be relaxed or the system must be redesigned to reduce the time required to perform the tasks in the sequence.

Step 5 in the job forming process requires identification of any constraints that might affect the partitioning of tasks into jobs. These constraints will restrict the formation of jobs and may arise due spatial considerations (i.e., distance between working areas in which tasks are performed) or a requirement that two or more tasks occur simultaneously or in rapid succession. Tasks that cannot be combined into the same job will be tagged to ensure that they are not combined. Another form of constraint is one that requires a set of tasks to be performed by the same person. Constraints of this type may cause tasks from different job tasks to be combined in the same job. The user will be asked if simultaneity or proximity constrains job function. The system default will be "no" constraints.

Step 6 of the process is at the heart of the job forming process. The process makes use of a network analysis technique know as the critical path method (CPM) or critical path scheduling (CPS). In the case of Category 1 tasks, the objective is to determine the number of jobs required to meet the mission timeline requirements for completing all the tasks required to successfully accomplish the function.

Step 6 is an iterative process through which tasks are assigned to a given crew size such that the response time is minimized. If the minimum response time achievable with a given crew size is unacceptable (i.e., it fails to meet the system requirement), the crew size will be increased. This process will continue until the point where either the requirement is met or further increases in crew size do not decrease the response time. This process is repeated for each job task containing Category 1 tasks. In each case, the minimum number of jobs that can still meet the required response times is determined. The largest of these minimum requirements is the lower bound for jobs for the weapon system for Category 1 tasks. If any of the functions must be carried out simultaneously, the number of jobs must increase to permit all of the required tasks to be completed within the required time for all functions that must be completed together.

Several slightly different algorithms are available for implementing the resource allocation process described above. Lang (1977) provides a heuristic approach for allocating a single type of resource to tasks in a critical path network. An algorithm for allocating multiple resources was

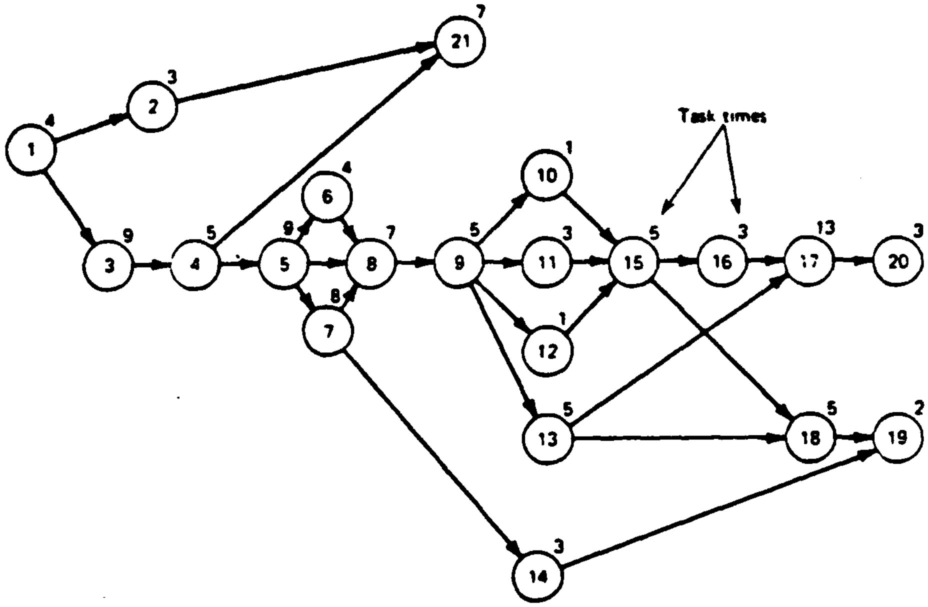


Figure B-2. Category 1 Tasks Arranged in a Precedence Network Based on Sequence and Time Requirements (from Bedworth and Bailey, 1982).

developed by Brooks (1963) and further extended by Bedworth (1973) and Bedworth and Bailey (1982). Bedworth and Bailey (1982) provided a computer coded algorithm that implements the Brook's algorithm.

Brook's (1963) algorithm (BAG) was selected for use in Product 5 for assigning Category 1 tasks to jobs. The computer coded algorithm is available for use in Product 5. The steps required to assign tasks (activities) to jobs (resources) are as follows. For convenience, Figure B-3 gives a network and tabular results of these steps based on three jobs.

A. Develop the task network, identifying tasks and their required times.

B. Determine the maximum time each task controls through the network on any one path. This is like calculating the critical-path time through the network assuming that the starting node for each task being analyzed is the network starting node. This activity control time will be designated ACTIM for convenience.

C. Rank these times in decreasing ACTIM sequence, as in Figure B-3 (G, A, C, etc.). ACTIM for task A is found by summing the times for tasks A, D, and E, to obtain a total of 16. The rows titled TEARL, TSTART, TFIN, and TNOW are explained as follows:

   1. TEARL is the earliest possible time, because of precedence and time limitations, to schedule each task. The actual time will be equal to or later than TEARL. TEARL equals the latest TFIN time for all immediate predecessor tasks.

   2. TSTART is the actual start time of the task. If there were no job limitations, TSTART would always equal TEARL.

   3. TFIN is the completion time of each task. This equals the tasks TSTART added to the job-duration time.

   4. TNOW is the time at which job assignments are now being considered. Initially TNOW equals zero, but subsequently it equals the lowest TFIN time for all tasks currently being worked on.

D. Sequence the tasks according to job constraints. TNOW is set at zero. The allowable tasks (ACT. ALLOW.) to be considered for scheduling at TNOW of zero are those tasks that would have a critical path method starting time of 0, namely tasks G, A, and C. These are placed in the ACT. ALLOW. row, sequenced in decreasing ACTIM order. In this example, G, A, and C all have the same ACTIM, and so a secondary rule is needed. For this example we will choose longest duration first, which dictates schedule G first. Another rule is needed for A and C, since both are five time-units long. Arbitrarily choose A before C. In the job-available column, the jobs initially available are placed--namely, three.

STEP A: Develop Task Network

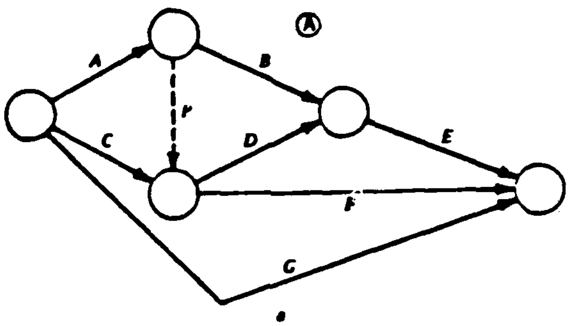
STEP B: Determine Task Duration

STEP C: Rank Tasks in Order of Network Durations (ACTIM)

STEP D: Determine Job Resources Available (3)

STEP E,F: Assign Tasks to Jobs, One Task at a Time in Network Order

STEP G: Repeat Process Until Jobs are Formed



Activity	Duration	Resources Required
A	5	1
B	4	1
C	5	1
D	7	1
E	4	1
F	8	1
G	16	1
P	Pseudo	—

ACTIVITY DATA	Activity	G	A	C	D	B	F	E
	Duration	16	5	5	7	4	8	4
	ACTIM C	16	16	16	11	8	8	4
	Resources required	1	1	1	1	1	1	1
	TEARL	0	0	0	5	5	5	12
	TSTART	0	0	0	5	5	9	12
	TFIN	16	5	5	12	9	17	16

PROJECT COMPLETION TIME

ALGORITHM ITERATION RESULTS					
TNOW		0	F 5	9	12
Resources available	D	3 2 1 0	2 1 0	1 0	1 0
ACT ALLOW	E	G A C	D B F	F	E
Iteration No.		1	2	3	4

G

Figure B-3. Brooks Algorithm Applied to Allocation of Category 1 Tasks to Multiple Jobs.

E. Determine if the first task in ACT. ALLOW., G, can be assigned. It can, since three jobs are available and G requires only one. Also, no predecessor limitations prevent G from beginning. G is removed from the ACT. ALLOW. list and the number of jobs available is decreased by one to a value of two, since G required one job. TSTART for task G is set at the current TNOW and the TFIN is set a TSTART plus task G's duration time. Now it is necessary to determine if task G being completed will allow another task to be feasible at some future time. With G it is not, since G is itself an entire critical path. This same process is repeated for the remainder of ACT. ALLOW. tasks until the jobs available are depleted. In this case, all task G, A, and C could be assigned a TSTART of zero. From the network of Figure 11 it is seen that assigning task A allows task B to be scheduled a TEARL of five time-units later (task A's TFIN). Similarly, tasks D and F can be assigned a TEARL that is the latest of A's and C's TFIN times. Note that if task A had required too many resources to allow assignment at TNOW of zero, we would still see if task C could be assigned.

F. TNOW is raised to the next TFIN time, which happens to be five, the completion times of both tasks A and C. The jobs available at TNOW of five is set to the number remaining after assigning resources at TNOW equal to zero (zero in this case), added to the number of jobs freed because of task completion at the new TNOW (two in this case). ACT. ALLOW. we now set at those not assigned at the previous TNOW (none in this case), added to those that have a TEARL equal to or less than TNOW (D, B, and F).

G. Repeat this assignment process until all tasks have been scheduled. The latest TFIN gives the response time that can be achieved with the resources assigned--in this case, 17 time units. Three jobs have been scheduled.

Step 7 in the job forming process provides a means for investigating alternative numbers of jobs and assessing the effect of these alternatives on the ability of the system to meet performance requirements. For example, a slight relaxation in the performance requirement might result in a need for one less job. Conversely, by adding another job to a weapon system, system performance may increase dramatically. Systems designers and Army decision makers need to be aware of such swings in both requirements and performance in order to make rational design decision.

The product of this process will be a listing of the unique jobs that result from Category 1 tasks. With each job will be a listing of the specific tasks associated with the job. Also, for each function consisting of Category 1 tasks, a resource profile will be shown that indicates what each job is required to do, over what time period, and the proportion of the soldier's time that is spent doing the tasks assigned to the job.

FROM BEDWORTH AND BAILEY (1982).

RESALL PROGRAM

```
FURPUR 27R3-U4    E33 SL73R1 03/25/80 09:23:35

IEE461*LIBRARY(1).RESALL
C**********************************************************************
C**********************************************************************
C**********************************************************************
C**********************************************************************
C
C                          THE RESALL PROGRAM
C
C                        (RESOURCE ALLOCATION)
C
C                    FOR FURTHER INFORMATION CONTACT
C                          DAVID D. BEDWORTH
C                   INDUSTRIAL ENGINEERING DEPARTMENT
C                       ARIZONA STATE UNIVERSITY
C                         TEMPE, ARIZONA 85281
C
C**********************************************************************
C**********************************************************************
C**********************************************************************
C
C          RESALL USES A HEURISTIC APPROACH TO ASSIGNING SCARCE
C          RESOURCES ON A CRITICAL PATH (CPM) TYPE PROJECT.
C            1. ASSIGNMENT OPTION: DETERMINES THE SHORTEST SCHEDULE
C               UNDER LIMITED MULTIPLE-RESOURCE CONSTRAINTS. A
C               MODIFICATION OF THE BROOKS ALGORITHM IS UTILIZED -
C               SEE PRODUCTION SYSTEMS CONTROL BY BEDWORTH AND BAILEY -
C               JOHN WILEY AND SONS - NEW YORK.
C
C            2. BALANCE OPTION: GIVEN A REQUIRED TIME SCHEDULE, THE
C               PROGRAM WILL DETERMINE THE MINIMUM AMOUNT OF A SINGLE
C               SCARCE RESOURCE TO ALLOW THE TIME SCHEDULE TO BE
C               MAINTAINED.AN ITERIVE APPROACH USING OPTION ALLOCATION
C               IS UTILIZED.
C
C          RESALL ALLOWS FOR BOTH NORMAL AND OVERTIME OPERATION AND
C          COSTING FOR AN ASSIGNMENT RUN ONLY. ONLY A NORMAL RUN IS
C          ALLOWED FOR A BALANCING PROBLEM.
C
C          RESALL WILL COMPUTE BASIC CPM DATA IF DESIRED BUT WILL NOT
C          PERFORM A CRASHING COST OPTIMIZATION.
C
C**********************************************************************
C**********************************************************************
C**********************************************************************
C**********************************************************************
C
```

```
C        INPUT REQUIREMENTS, USING CARD FORMATS, ARE AS FOLLOWS:
C        CARD J REPRESENTS TYPE J. THERE SHOULD BE MORE THAN 1 CARD OF
C        TYPES 3 AND 8 FOR THIS PROGRAM.
C         (ALL INTEGER-FORMAT DATA IS RIGHT JUSTIFIED IN FIELD)
C        CARD 1, COLS 1 - 78: FORMAT 20A4, USER TITLE INFORMATION.
C        CARD 2, COLS 1 - 10: TOTAL NUMBER OF ACTIVITIES IN THE
C                             PROJECT, BETWEEN 3 AND 100 INCLUSIVE,
C                             FORMAT F10.0.
C                COLS 11- 20: TOTAL NUMBER OF RESOURCE TYPES,
C                             BETWEEN 1 AND 20 INCLUSIVE, F10.0.
C                COLS 21- 30: CRITICAL PATH TIME FROM CPM PROGRAM
C                             RUN. IF 0. IS INPUT, CPM DATA WILL
C                             BE COMPUTED.
C                COLS 31- 40: FIXED (INDIRECT) COST PER TIME PERIOD,
C                             F10.0. THIS MAY BE 0. FOR NO COSTING.
C                COLS 41- 50: STARTING TIME FOR NORMAL SCHEDULE
C                             (USUALLY 0.), F10.0.
C                COLS 51- 60: STARTING TIME FOR OVERTIME SCHEDULE
C                             (USUALLY 0.), F10.0.
C                COL  61:     IF 0, INHIBITS DETAILED PRINTING FOR
C                             EACH ITERATION. IF 1, GIVES DETAILED
C                             PRINTING.
C                COL  62:     IF 0, INHIBITS PRINTING OF FINAL
C                             RESOURCE PROFILES OVER TIME. IF 1,
C                             ALLOWS RESOURCE PROFILE PRINTING.
C                COLS 63- 67: IF 0, THIS IS ASSIGNMENT OPTION. IF
C                             1., BALANCE - F5.0.
C                COLS 68- 77: TIME REQUIRED FOR PROJECT BALANCE
C                             RUN - F10.0.
C
C        CARD 3+, COLS. 1-32, FORMAT 8A4, DESCRIPTION OF RESOURCE
C                 - ONE CARD NEEDED FOR EACH RESOURCE - PUT IN SAME
C                 SEQUENCE AS GIVEN ON TIME AND COST CARDS
C
C        CARD 4, NORMAL RESOURCE QUANTITIES, 20I4 FORMAT. RESOURCE
C                 1 IN COLS 1-4; RESOURCE 2 IN COLS 5-8 ETC. LEAVE
C                             NON-USED RIGHT-JUSTIFIED RESOURCE
C                             COLUMNS BLANK.
C
C        CARD 5, OVERTIME RESOURCE QUANTITIES - SAME PROCESS AS
C                 FOR CARD 4 - FORMAT IS 20I4.
C
C        CARD 6, NORMAL RESOURCE COSTS, 20I4 - SAME PROCESS AS
C                 FOR CARD 1. THESE ARE DIRECT COSTS.
C
C        CARD 7, OVERTIME RESOURCE COSTS, 20I4 -  SAME PROCESS AS
C                 FOR CARD 4. THESE ARE DIRECT COSTS.
C
C        CARD 8+, ACTIVITY CARDS - ONE PER ACTIVITY:
C                 COLS 1 -  3: TAIL NODE NUMBER, I3.
C                 COLS 4 -  6: HEAD NODE NUMBER, I3.
C                 COLS 7 -  9: EARLIEST START TIME FOR ACTIVITY
                               - NOT NEEDED IF CRITICAL PATH
C                                FOUND IN THIS RUN. (I3)
C                   COLS 10- 12: DURATION TIME FOR ACTIVITY, I3.
C                   COLS 13- 15: TOTAL FLOAT FOR ACTIVITY, I3. NOT
C                                NEEDED IF CRITICAL PATH FOUND IN
C                                THIS RUN.
C                   COLS 16- 18: FREE FLOAT FOR ACTIVITY, I3. NOT
C                                NEEDED IF CRITICAL PATH FOUND IN
C                                THIS RUN.
C                   COLS 19- 78: RESOURCE QUANTITIES NEEDED FOR THIS
C                                ACTIVITY, 20I3. RESOURCE 1 IN COLS
C                                19-21, RESOURCE 2 IN COLS 22-24,
C                                ETC. UNUSED RIGHT-JUSTIFIED
C                                RESOURCES LEAVE BLANK.
```

```
C
C          ANOTHER SETUP MAY FOLLOW. TWO BLANK CARDS SIGNIFY LAST
C          SETUP
C          AN EXAMPLE OF DECK SET-UP FOLLOWS WHERE USER DOES NOT
C          FURNISH DATA FROM CPM RUN. TWO SCARCE RESOURCES ANALYZED:
C
C
C      RESALL ASSIGNMENT RUN - TEST.
C        5.        2.        0.        0.        0.        0.00   0.
C      RESOURCE 1
C      RESOURCE 2
C  3   4
C  2   2
C 25  30
C 40  60
C 1  2  0  5  0  0  2  3
C 2  4  0  7  0  0  1  1
C 1  3  0  3  0  0  3  2
C 3  4  0 10  0  0  2  0
C
C
C*******************************************************************
C*******************************************************************
C*******************************************************************
C
C          RESALL WAS COMPILED IN ASCII-FORTRAN ON A UNIVAC - 1110.
C
C*******************************************************************
C*******************************************************************
C*******************************************************************
C*******************************************************************
C          BOUNDS ON RESALL PROBLEMS INCLUDE:
C           1. A MINIMUM OF 3 PROJECT ACTIVITIES AND A MAXIMUM OF 100.
C           2. A MINIMUM OF 1 RESOURCE AND A MAXIMUM OF 20.
C           3. ONLY 1 RESOURCE ALLOWED FOR A BALANCE RUN.
C           4. NO OVERTIME CONDITIONS ALLOWED FOR A BALANCE RUN.
C*******************************************************************
C*******************************************************************
C*******************************************************************
C*******************************************************************
C
C            RESALL WAS ORIGINALLY DEVELOPED BY RICHARD MASON, WORKING
C            WITH DAVID BEDWORTH, FOR PART OF THE MS RESEARCH PAPER
C            TITLED AN ADAPTATION OF THE BROOKS ALGORITHM FOR SCHEDULING
C            PROJECTS UNDER MULTIPLE RESOURCE CONSTRAINTS - INDUSTRIAL
C            ENGINEERING DEPARTMENT, ARIZONA STATE UNIVERSITY, TEMPE:1970
C            SUBSEQUENT CHANGES AND ADDITIONS WERE MADE BY DAVID BEDWORTH
C
C*******************************************************************
C*******************************************************************
C*******************************************************************
C*******************************************************************
      CHARACTER*6 TITLE(13), RTIT(20,30)
      DIMENSION ESX(100), TEMP(100), NRPL(100), POOL(22), USED(22), SP(2
     12), IPOOL(22), IUSED(22), ISP(22), IT(100), IN(100), IASST(100), K
     2RES(20), KORES(20), KCSTN(20), LINE(110), NRCEN(200,20), KCSTO(20)
     3, NRCEO(200,20), IASFT(100), ITEMP(100)
      COMMON T(100), N(100), DUR(100), ES(100), TF(100), FF(100), CPTD,
     1SNORM, NACT, KEXIT, ASST(100), ASFT(100), RN(100,20), NRES
      INTEGER OVERT, BLANK, RES(20), ORES(20), CSTN(20), CSTO(20), T, N,
     1 ES, DUR, TF, FF, RN
      DATA NORM/1HN/, OVERT/1HO/, BLANK/1H /
C
C*****READ PROJECT HEADER - REQUIRED
C
    5 READ 10,TITLE
```

```
   10 FORMAT (13A6)
C
C*****READ PARAMETER CARD--TOTAL NUMBER OF ACTIVITIES, TOTAL
C*****TYPES OF RESOURCES, CRITICAL PATH TERMINATION DATE,
C*****FIXED COST RATE, START TIME OF NORMAL SCHEDULE, AND
C*****START TIME OF OVERTIME SCHEDULE.
C
      READ 15,TACT,TRES,CPTD,FIX,SNORM,SOVER,IPPD,IRSN,BALNC,TREQ
   15 FORMAT (6F10.0,2I1,F5.0,F10.0)
      IF (TACT.EQ.0.) CALL EXIT
      NACT=TACT
      NRES=TRES
      OUT=0.
      IVERTP=0
      KEXIT=1
C
C*****READ RESOURCE HEADERS - ONE REQUIRED/RESOURCE
C
      DO 20 I=1,NRES
      READ 25,(RTIT(I,J),J=1,5)
   20 CONTINUE
   25 FORMAT (5A6)
C
C*****READ NORMAL AND OVERTIME RESOURCE QUANTITITY CARDS,
C
      READ 30,(RES(I),I=1,20)
      READ 30,(ORES(J),J=1,20)
   30 FORMAT (20I4)
C
C*****READ NORMAL AND OVERTIME RESOURCE COST CARDS.
C
      READ 30,(CSTN(J),J=1,20)
      READ 30,(CSTO(K),K=1,20)
      DO 35 I=1,20
      KRES(I)=RES(I)
      KORES(I)=ORES(I)
      KCSTN(I)=CSTN(I)
      KCSTO(I)=CSTO(I)
   35 CONTINUE
C
C*****INITIALIZE PERIOD RESOURCE VALUES.
C
      DO 40 I=1,200
      DO 40 J=1,NRES
      NRCEN(I,J)=0
   40 NRCEO(I,J)=0
C
C*****READ ACTIVITY CARDS--TAIL, HEAD, EARLIEST START TIME,
C*****DURATION, TOTAL FLOAT, FREE FLOAT--ALL DETERMINED BY
C*****A PREVIOUS CPM ANALYSIS,
C
      DO 45 I=1,NACT
   45 READ 50,T(I),H(I),ES(I),DUR(I),TF(I),FF(I),(RN(I,J),J=1,NRES)
   50 FORMAT (26I3)
C
C*****PRINT INPUT INFORMATION
C
      INFLAG=1
      IF (CPTD.NE.0.) GO TO 55
      INFLAG=0
      CALL CRITIC
   55 PRINT 60
   60 FORMAT (1H1,2X,67HTHIS IS THE BEDWORTH-MASON ADAPTATION OF BROOKS
     1RESOURCE ALGORITHM.,///,2X,29HTHE DATA INPUT IS AS FOLLOWS:,///)
      PRINT 65,TITLE
   65 FORMAT (//,2X,14HPROJECT TITLE:,13A6,//)
```

```
      IF (INFLAG.EQ.0) GO TO 75
      PRINT 70
   70 FORMAT (//,4X,30HCRITICAL PATH AND FLOATS INPUT BY USER,///)
      GO TO 85
   75 PRINT 80
   80 FORMAT (//,4X,42HCRITICAL PATH AND FLOATS NOT INPUT BY USER,///)
   85 PRINT 90,TACT
   90 FORMAT (//,4X,21HNUMBER OF ACTIVITIES:,F14.0)
      PRINT 95,TRES
   95 FORMAT (//,4X,25HNUMBER OF RESOURCE TYPES:,F10.0)
      PRINT 100,CPTD
  100 FORMAT (//,4X,19HCRITICAL PATH TIME:,F16.0)
      PRINT 105,FIX
  105 FORMAT (//,4X,25HFIXED COST PER TIME UNIT:,F10.0)
      PRINT 110,SNORM
  110 FORMAT (//,4X,26HSTART TIME (NORMAL SCHED):,F9.0)
      PRINT 115,SOVER
  115 FORMAT (//,4X,26HSTART TIME (OVERTIME SCH):,F9.0)
      IF (BALNC.GT.0.) GO TO 125
      PRINT 120
  120 FORMAT (//,4X,26HTHIS IS AN ALLOCATION RUN.)
      GO TO 135
  125 PRINT 130,TREQ
  130 FORMAT (//,4X,43HTHIS IS A BALANCING RUN. TIME REQUESTED IS:,F5.0)
  135 PRINT 140
  140 FORMAT (//,4X,35HRESOURCE INFORMATION IS AS FOLLOWS:,/)
      PRINT 145
  145 FORMAT (/,4X,8HRESOURCE,35X,8HQUANTITY,19X,4HCOST)
      PRINT 150
  150 FORMAT (2X,13HNUMBER  TITLE,30X,15HNORMAL OVERTIME,9X,15HNORMAL OV
     1ERTIME,/)
      DO 155 I=1,NRES
  155 PRINT 160,I,(RTIT(I,J),J=1,5),RES(I),ORES(I),CSTN(I),CSTO(I)
  160 FORMAT (4X,I3,3X,5A6,3X,I5,5X,I5,10X,I5,4X,I5)
      PRINT 265
      PRINT 165
  165 FORMAT (//,3X,21HACTIVITY INFORMATION:,/)
      PRINT 170
  170 FORMAT (/,1X,14HIDENTIFICATION,1X,6HEARLY.,5X,5HTOTAL,2X,4HFREE,20
     1X,18HRESOURCES REQUIRED)
      PRINT 175
  175 FORMAT (4X,9HTAIL-HEAD,3X,5HSTART,1X,4HTIME,1X,5HFLOAT,1X,86HFLOAT
     1  R1  R2  R3  R4  R5  R6  R7  R8  R9  R10 R11 R12 R13 R14 R15 R16
     2R17 R18 R19 R20,/)
      DO 180 I=1,NACT
  180 PRINT 185,T(I),H(I),ES(I),DUR(I),TF(I),FF(I),(RN(I,J),J=1,NRES)
  185 FORMAT (4X,I3,2X,I3,4X,I3,3X,I3,3X,I3,3X,I3,1X,9I4,1X,11I4)
      IF (NACT.LT.3) GO TO 960
C
C*****TEST FOR RESTRICTIONS ON RESOURCE BALANCING
C
      IF (BALNC.EQ.0.) GO TO 195
      IF (NRES.GT.1) GO TO 940
      IF (CPTD.GT.TREQ) GO TO 950
C
C*****DETERMINE STARTING RESOURCE LEVEL - MAXIMUM FOR ANY ACTIVITY
C
      MAXR=0
      DO 190 I=1,NACT
      IF (RN(I,1).GT.MAXR) MAXR=RN(I,1)
  190 CONTINUE
      RES(1)=MAXR
      KCSTO(1)=0
      ORES(1)=0.
      KORES(1)=0
      KRES(1)=RES(1)
      JRES=KRES(1)
```

```
C
C*****TEST FOR PROJECT FEASIBILITY.  DETERMINE IF THE RESOURCE
C*****REQUIREMENTS FOR ANY ACTIVITY EXCEED THOSE AVAILABLE.
C
  195 DO 200 M=1,NRES
  200 POOL(M)=RES(M)+ORES(M)
      DO 205 I=1,NACT
      DO 205 J=1,NRES
      IF (POOL(J)-RM(I,J)) 775,205,205
  205 CONTINUE
C
C*****COMPUTE MAXIMUM REMAINING PATH PENGTHS.
C
      DO 210 I=1,NACT
      SUM=ES(I)+TF(I)
  210 MRPL(I)=CPTD-SUM
C
C*****ARRANGE ACTIVITY DATA IN ORDER OF LONGEST REMAINING PATH
C*****LENGTH,  BREAK TIES BY RANKING THE ACTIVITY WITH THE
C*****LONGEST DURATION FIRST.
C
  215 KRUD=0
      NL1=NACT-1
      DO 240 I=1,NL1
      IP1=I+1
      DO 235 J=IP1,NACT
      IF (MRPL(I)-MRPL(J)) 225,220,235
  220 IF (DUR(I).GE.DUR(J)) GO TO 235
      KRUD=1
  225 DO 230 L=1,NRES
      TEMP(L)=RM(I,L)
      RM(I,L)=RM(J,L)
  230 RM(J,L)=TEMP(L)
      SORT=MRPL(I)
      MRPL(I)=MRPL(J)
      MRPL(J)=SORT
      SORT=T(I)
      T(I)=T(J)
      T(J)=SORT
      SORT=N(I)
      N(I)=N(J)
      N(J)=SORT
      SORT=ES(I)
      ES(I)=ES(J)
      ES(J)=SORT
      SORT=DUR(I)
      DUR(I)=DUR(J)
      DUR(J)=SORT
      SORT=FF(I)
      FF(I)=FF(J)
      FF(J)=SORT
      SORT=TF(I)
      TF(I)=TF(J)
      TF(J)=SORT
  235 CONTINUE
  240 CONTINUE
      IF (KRUD) 245,245,215
C
C*****INITIALIZE TEMPORARY STORAGE LOCATIONS.
C
  245 PTIME=SHORR
      IF (OUT.GT.0.0) PTIME=SOVER
      CORR=0.0
      TIME=0.
      OUT=0.
      IACT=0.
      PSFT=0.
```

```
      SUPP=0.
      OVER=0.
      TOT=0.
      COST=0.
      CNORM=0.
      TFIX=0.
      TIDLE=0.
      TNORM=0.
      TOVER=0.
      TOTAL=0.
      KOUNT=6
      DO 250 J=1,22
  250 SP(J)=0.
      DO 255 I=1,NACT
      ESX(I)=ES(I)
      ASFT(I)=0.
      TEMP(I)=0.
  255 USED(I)=0.
C
C*****COMPUTE THE COST OF NORMAL AND OVERTIME RESOURCES.
C*****PRINT APPROPRIATE OUTPUT TITLES.
C
      DO 260 N=1,NRES
      CN=CSTN(N)*RES(N)
      CO=CSTO(N)*ORES(N)
      OUT=OUT+CO
  260 COST=COST+CN
      PRINT 265
  265 FORMAT (1H1)
      IF (BALNC.GT.0.) GO TO 270
      IF (IPPD.NE.1) GO TO 345
      PRINT 275
      GO TO 290
  270 PRINT 280
      PRINT 285
  275 FORMAT (/,56X,25HTHIS IS AN ALLOCATION RUN)
  280 FORMAT (/,56X,23HTHIS IS A BALANCING RUN)
  285 FORMAT (///,4X,91HOVERTIME RESOURCES ARE OF NO BENEFIT IN A BALANC
     1ING RUN.  IF AN OVERTIME QUANTITY WAS INPUT,/,4X,68HBY THE USER IT
     2 WILL BE SET TO ZERO BY THIS PROGRAM BEFORE BALANCING.,//)
  290 IF (OUT) 305,305,295
  295 PRINT 300
  300 FORMAT (59X,17HOVERTIME SCHEDULE,/)
      GO TO 325
  305 PRINT 310
  310 FORMAT (60X,15HNORMAL SCHEDULE,/)
      IF (BALNC.EQ.0.) GO TO 325
      PRINT 315,JRES
  315 FORMAT (/,10X,27HSTARTING RESOURCE LEVEL IS ,I3,6H UNITS)
      KREQ=TREQ
      PRINT 320,KREQ
  320 FORMAT (/,10X,26HMAXIMUM TIME REQUESTED IS ,I4,11H TIME UNITS)
      GO TO 360
  325 PRINT 330
  330 FORMAT (46X,35HPERIOD RESOURCES CONSUMED AND COSTS,8H SUMMARY,/)
      PRINT 335
  335 FORMAT (7X,8HACTIVITY,30X,15HRESOURCE VALUES)
      PRINT 340
  340 FORMAT (2X,11HPERIOD SLIP,2X,110HR1  R2  R3  R4  R5  R6  R7  R8  R
     19  R10 R11 R12 R13 R14 R15 R16 R17 R18 R19 R20    FIXED    IDLE  N
     2ORMAL   OVER  TOTAL)
  345 DO 350 J=1,20
  350 IPOOL(J)=POOL(J)
      IF (IPPD.NE.1) GO TO 360
      PRINT 355,(IPOOL(J),J=1,20)
  355 FORMAT (2X,11HQUANTITY: (,20I4,1H),//)
```

```
C
C*****COMPUTE FIXED COSTS, AND IDLE, NORMAL, AND OVERTIME
C*****RESOURCE COSTS, AND SUM TO FIND THE TOTAL COST FOR A
C*****GIVEN TIME PERIOD.
C
  360 DO 365 L=1,1000
      IF (IACT) 455,455,365
  365 DO 380 I=1,NRES
      USED(I)=RES(I)+DRES(I)-POOL(I)
      OT=USED(I)-RES(I)
      IF (OT) 375,380,370
  370 COT=CSTO(I)*OT
      OVER=OVER+COT
      GO TO 380
  375 UU=RES(I)-USED(I)
      SP(1)=CSTN(I)*UU
      SUPP=SUPP+SP(1)
  380 CONTINUE
      CNORM=COST-SUPP
      IF (NACT-IACT) 385,385,390
  385 IF (PSFT-TIME) 570,390,390
  390 TOT=FIX+SUPP+CNORM+OVER
      IF (BALNC.GT.0.) GO TO 400
      IF (IPPD.NE.1) GO TO 400
      IF (KOUNT-20) 400,400,395
C
C*****PRINT FOR THE CURRENT TIME PERIOD THE RESOURCES CONSUMED
C*****AND COST SUMMARY INFORMATION.
C
  395 PRINT 500
      PRINT 265
      PRINT 340
      KOUNT=0
  400 DO 415 J=1,NRES
      IUSED(J)=USED(J)
      IF (OUT) 405,405,410
  405 NRCEN(L,J)=IUSED(J)
      GO TO 415
  410 NRCED(L,J)=IUSED(J)
  415 CONTINUE
      ITIME=PTIME
      JTIME=L
      IF (BALNC.GT.0.) GO TO 430
      IF (IPPD.NE.1) GO TO 430
      PRINT 420,ITIME,(IUSED(J),J=1,NRES)
  420 FORMAT (/,1H+,I3,9H  *******,20I4)
      PRINT 425,FIX,SUPP,CNORM,OVER,TOT
  425 FORMAT (71X,26HCOSTS FOR THIS PERIOD ARE:,5F7.0,/)
      KOUNT=KOUNT+2
  430 TFIX=TFIX+FIX
      TIDLE=TIDLE+SUPP
      TNORM=TNORM+CNORM
      TOVER=TOVER+OVER
      TOTAL=TOTAL+TOT
      SUPP=0.
      CNORM=0.
      OVER=0.
      TOT=0.
C
C*****DETERMINE IF ANY ACTIVITIES END AT THE END OF THE
C*****CURRENT TIME PERIOD.  IF SO, PLACE THEIR RESOURCES BACK
C*****INTO THE POOL.
C
      DO 450 I=1,NACT
      ETIME=ASFT(I)-CORR
      IF (TIME-ETIME) 450,435,450
```

```
  435 IF (ASFT(I)) 450,450,440
  440 DO 445 J=1,NRES
      POOL(J)=POOL(J)+RM(I,J)
  445 CONTINUE
  450 CONTINUE
C
C*****DETERMINE IF ANY ACTIVITIES ARE SCHEDULED TO START AT
C*****THE END OF THE CURRENT TIME PERIOD.  IF SO, DETERMINE
C*****THE ORDER IN WHICH RESOURCES ARE TO BE ALLOCATED.
C*****PRIORITY IS GIVEN TO THE ACTIVITY WITH THE LONGEST
C*****REMAINING PATH LENGTH.  IF THERE ARE INSUFFICIENT
C*****RESOURCES, SLIP THE ACTIVITY ONE TIME UNIT FOR
C*****CONSIDERATION IN THE NEXT TIME PERIOD.
C
  455 DO 535 I=1,NACT
      SHORT=0.
      IF (TIME-ESX(I)) 535,460,460
  460 DO 470 J=1,NRES
      POOL(J)=POOL(J)-RM(I,J)
      IF (POOL(J)) 465,470,470
  465 SHORT=SHORT+POOL(J)
  470 CONTINUE
      IF (SHORT) 475,520,520
  475 DO 480 N=1,NRES
      SP(N)=POOL(N)
  480 POOL(N)=POOL(N)+RM(I,N)
      ESX(I)=ESX(I)+1.
      DO 490 NN=1,NRES
      IF (SP(NN)) 490,490,485
  485 SP(NN)=0.
  490 CONTINUE
      IF (BALNC.GT.0.) GO TO 505
      IF (IPPD.NE.1) GO TO 505
      IF (KOUNT-20) 505,505,495
  495 PRINT 500
  500 FORMAT (5X,26HFOR THE ABOVE INFORMATION:,/,3X,101H1. ******* FOR A
     1CTIVITY SLIP MEANS THAT THE FOLLOWING RESOURCES WERE CONSUMED DURI
     2NGTHAT TIME PERIOD.,/,3X,107H2. WHERE ACTIVITY IS LISTED UNDER ACT
     3IVITY SLIP, IT MEANS THAT ACTIVITY COULD NOT BE SCHEDULED IN THAT
     4TIME,/,5X,93HPERIOD BECAUSE OF A RESOURCE SHORTAGE. THE RESOURCE C
     5AN BE IDENTIFIED BY A NEGATIVE QUANTITY.)
      PRINT 265
      PRINT 340
      KOUNT=0
C
C*****PRINT THE ELIGIBLE ACTIVITY THAT HAS BEEN SLIPPED DUE TO
C*****A LACK OF RESOURCES AND IDENTIFY THE RESOURCE SHORTAGES,
C
  505 TYME=PTIME+1.
      ITYME=TYME
      IT(I)=T(I)
      IN(I)=N(I)
      DO 510 J=1,NRES
  510 ISP(J)=SP(J)
      IF (BALNC.GT.0.) GO TO 530
      IF (IPPD.NE.1) GO TO 530
      PRINT 515,ITYME,IT(I),IN(I),(ISP(NN),NN=1,NRES)
  515 FORMAT (/,1X,I3,1X,I3,2H -,I3,20I4)
      KOUNT=KOUNT+1
      GO TO 530
C
C*****SCHEDULE THE ACTIVITY, AND THEN REMOVE IT FROM FUTURE
C*****CONSIDERATION.  COUNT ACTIVITIES SCHEDULED.  COMPUTE
C*****CURRENT LATEST PROJECT SHCEDULED FINISH TIME.
C
  520 ASST(I)=PTIME
```

```
      ASFT(I)=PTIME+DUR(I)
      ES(I)=ES(I)+9999.
      ESX(I)=ES(I)
      IACT=IACT+1
      IF (ASFT(I)-PSFT) 530,530,525
  525 PSFT=ASFT(I)
  530 TEMP(I)=ASFT(I)
  535 CONTINUE
C
C***** TEST ACTIVITIES TO SEE IF SLIPPAGE CAN BE ABSORBED WITH
C*****FREE FLOAT,  IF NOT, COMPUTE ACTIVITY DELAY, AND THEN
C*****ADJUST ACCORDINGLY THE EARLIEST START TIME OF THE
C*****ACTIVITY AND THE EARLIEST START TIMES OF ANY IMMEDIATE
C*****DEPENDENT ACTIVITIES.
C
      DO 560 I=1,NACT
      IF (ES(I)-ESX(I)) 540,560,560
  540 DIFF=ESX(I)-ES(I)
      IF (FF(I)-DIFF) 545,560,560
  545 DLAY=DIFF-FF(I)
      HEAD=H(I)
      DO 555 K=1,NACT
      IF (T(K)-HEAD) 555,550,555
  550 ESX(K)=ES(K)+DLAY
  555 CONTINUE
  560 CONTINUE
      TIME=TIME+1.
      IF (IACT.GT.0) PTIME=PTIME+1.
      IF (IACT.GT.0) CORR=PTIME-TIME
  565 CONTINUE
      IF (BALNC.GT.0.) GO TO 605
      IF (IPPD.NE.1) GO TO 575
      IF (KOUNT.GT.0) PRINT 500
      IF (KOUNT-20) 590,590,575
  570 IF (BALNC.GT.0.) GO TO 580
      IF (IPPD.NE.1) GO TO 575
      IF (KOUNT.GT.0) PRINT 500
      IF (KOUNT-16) 590,590,575
  575 PRINT 265
      GO TO 590
  580 PRINT 585,KRES(1)
  585 FORMAT (/,10X,39HRESOURCE FOR THIS ITERATION WAS SET AT ,I3,7H UNI
     1TS.)
  590 IF (OUT) 620,620,595
  595 PRINT 600
  600 FORMAT (////)
      PRINT 300
      GO TO 630
  605 PRINT 265
      PRINT 610,TITLE
  610 FORMAT (13A6,//)
      PRINT 615,KRES(1)
  615 FORMAT (/,10X,39HRESOURCE FOR THIS ITERATION WAS SET AT ,I3,7H UNI
     1TS.)
      GO TO 655
  620 PRINT 625
  625 FORMAT (////)
      PRINT 310
  630 PRINT 635
  635 FORMAT (//,50X,19HTOTAL PROJECT COSTS,//)
      PRINT 640
  640 FORMAT (35X,5HFIXED,11X,4HIDLE,9X,6HNORMAL,7X,8HOVERTIME,10X,5HTOT
     1AL,/)
      PRINT 645,TFIX,TIDLE,TNORM,TOVER,TOTAL
  645 FORMAT (25X,5F15.0)
      PRINT 650
```

```
  650 FORMAT (//,35X,66HA DETAILED SCHEDULE FOR THIS RUN IS GIVEN ON THE
     1 NEXT OUTPUT PAGE.)
C
C*****PRINT PROJECT ACTIVITY SCHEDULE, WHICH CONSISTS OF A
C*****LISTING OF THE REVISED ACTIVITY START AND FINISH TIMES.
C
  655 PRINT 265
      KKK=13
      IF (OUT) 670,670,660
  660 PRINT 665
  665 FORMAT (//,28X,17HOVERTIME SCHEDULE,/)
      GO TO 680
  670 PRINT 675
  675 FORMAT (//,29X,15HNORMAL SCHEDULE,/)
  680 PRINT 685
  685 FORMAT (//,24X,26H PROJECT ACTIVITY SCHEDULE,/)
      ICPTD=CPTD
      PRINT 690,ICPTD
  690 FORMAT (19X,30HCRITICAL PATH TERMINATION DATE,I5,///)
      PRINT 695
  695 FORMAT (14X,38HTHESE ARE THE REVISED START AND FINISH,6H TIMES,/)
      PRINT 700
  700 FORMAT (/,21X,32H ACTIVITY      START      FINISH)
      PRINT 705
  705 FORMAT (20X,10HTAIL  HEAD,8X,15H(END OF PERIOD),//)
C
C*****USE CRITIC SUBROUTINE TO REORDER ACTIVITIES
C*****PRIOR TO PRINTING THE FINAL SCHEDULE.
C
      KEXIT=0
      CALL CRITIC
      DO 720 I=1,NACT
      KKK=KKK+1
      IF (KKK-53) 715,715,710
  710 PRINT 265
      PRINT 700
      PRINT 705
      KKK=0
  715 IT(I)=T(I)
      IH(I)=H(I)
      IASST(I)=ASST(I)
      IASFT(I)=ASFT(I)
  720 PRINT 725,IT(I),IH(I),IASST(I),IASFT(I)
  725 FORMAT (20X,I3,3H - ,I3,7X,I3,9X,I3)
      DO 735 I=1,NL1
      IP1=I+1
      DO 735 J=IP1,NACT
      IF (TEMP(I)-TEMP(J)) 730,735,735
  730 SORT=TEMP(I)
      TEMP(I)=TEMP(J)
      TEMP(J)=SORT
  735 CONTINUE
      ITEMP(1)=TEMP(1)
      IF (OUT.GT.0.) IVERTP=ITEMP(1)
      PRINT 740,ITEMP(1)
  740 FORMAT (//,15X,27HMINIMUM PROJECT DURATION = ,I3,11H TIME UNITS)
      IF (BALNC.GT.0.) GO TO 790
C
C*****TEST TO DETERMINE IF THE SCHEDULE JUST PRINTED WAS AN
C*****OVERTIME SCHEDULE.  IF SO, ZERO THE OVERTIME RESOURCE
C*****AND COST ARRAYS, AND THEN REPEAT THE PROGRAM TO COMPUTE
C*****THE NORMAL SCHEDULE.
C
  745 IF (OUT) 785,785,750
  750 IF (BALNC.EQ.0.) GO TO 755
      KORESL=KRES(1)
```

```
      KRES(1)=JRES-ORES(1)
      JRES=KRES(1)
  755 DO 760 KO=1,NRES
      ORES(KO)=0.
  760 CSTO(KO)=0.
  765 DO 770 NO=1,NACT
  770 ES(NO)=ES(NO)-0.
      GO TO 195
  775 PRINT 780
  780 FORMAT (///,10X,98HPROJECT COULD NOT BE COMPLETED AS RESOURCES REQ
     1UIRED EXCEEDED AVAILABLE RESOURCES ON THE NEXT RUN.)
      GO TO 5
  785 IF (BALNC.GT.0.) GO TO 795
      GO TO 810
  790 IF (ITEMP(1).GT.TREQ) GO TO 805
      GO TO 745
  795 KNRESL=KRES(1)
      PRINT 800
  800 FORMAT (//,50X,17HBALANCE COMPLETE.)
      GO TO 810
  805 RES(1)=KRES(1)+1.0
      KRES(1)=RES(1)
      GO TO 765
C
C*****PRINT RESOURCE  UTILIZATION PROFILES
C
  810 K=1
      IF (IRSM.NE.1) GO TO 5
      KTIME=JTIME
      JTIME=JTIME-1
  815 PRINT 820,K
  820 FORMAT (1H1,50X,15HRESOURCE NUMBER,1X,I2,1X,7HSUMMARY,//)
      PRINT 825,(RTIT(K,J),J=1,5)
  825 FORMAT (20X,16HRESOURCE ITEM IS,1X,5A6)
      IF (BALNC.GT.0.) GO TO 830
      KTOT=KRES(K)+KORES(K)
      GO TO 835
  830 KTOT=0
      KRES(1)=KNRESL
  835 PRINT 840,KRES(K),KTOT
  840 FORMAT (20X,23HNORMAL QUANTITY IS:    ,I3,20X,21HOVERTIME QUANTITY
     1 IS:,I3,23H. (NORMAL AND OVERTIME))
      PRINT 845,KCSTN(K),KCSTO(K)
  845 FORMAT (20X,23HNORMAL COST/PERIOD IS: ,I3,20X,21HOVERTIME COST/PER
     1IOD:,I3,/)
      PRINT 850
  850 FORMAT (52X,23HUTILIZATION INFORMATION,/)
      PRINT 855
  855 FORMAT (20X,15HNORMAL SCHEDULE,25X,17HOVERTIME SCHEDULE)
      PRINT 860
  860 FORMAT (14X,8HQUANTITY,6X,19HPERCENT UTILIZATION,9X,8HQUANTITY,6X,
     119HPERCENT UTILIZATION)
      PRINT 865
  865 FORMAT (5X,4HTIME,7X,4HUSED,7X,22H0 1 2 3 4 5 6 7 8 9 10,9X,4HUSED
     1,7X,22H0 1 2 3 4 5 6 7 8 9 10,//)
      DO 870 N=1,110
  870 LINE(N)=BLANK
      DO 930 L=2,KTIME
      IPT=(((NRCEN(L,K))*21.)/(KRES(K)))+20.
      JPT=((NRCEO(L,K))*21.)/(KRES(K)+KORES(K))+70.
      DO 875 N=20,IPT
  875 LINE(N)=NORM
      KFLAG=0
      DO 890 N=70,JPT
      IF (L-IVERTP-1) 880,880,885
  880 LINE(N)=OVERT
```

```
      GO TO 890
  885 KFLAG=1
  890 CONTINUE
      LT=L-1
      IF (KFLAG) 895,895,905
  895 PRINT 900,LT,NRCEN(L,K),(LINE(I),I=28,49),NRCEO(L,K),(LINE(I),I=70
     1,91)
  900 FORMAT (5X,I3,8X,I3,8X,22A1,8X,I3,9X,22A1)
      GO TO 915
  905 PRINT 910,LT,NRCEN(L,K),(LINE(I),I=28,49),(LINE(I),I=70,91)
  910 FORMAT (5X,I3,8X,I3,8X,22A1,20X,22A1)
  915 DO 920 N=28,IPT
  920 LINE(N)=BLANK
      DO 925 N=70,JPT
  925 LINE(N)=BLANK
  930 CONTINUE
      PRINT 935
  935 FORMAT (//,58X,39HPARTIAL OVERTIME UTILIZATION ASSUMES NORMAL QUAN
     1TITIES USED,/,58X,72HFULLY BEFORE OVERTIME FOR COSTING. FOR EXAMPL
     2E, IF THE OVERTIME QUANTITY,/,58X,60H(NORMAL AND OVERTIME) IS 2 AN
     3D ONLY 1 UNIT IS USED, THEN THE,/,58X,53HUTILIZATION COST IS NORMA
     4L AND THE IDLE COST IS ZERO.)
      K=K+1
      IF (NRES.GE.K) GO TO 815
      GO TO 5
  940 PRINT 945
  945 FORMAT (1H0,2X,73HNO BALANCE AS NUMBER OF RESOURCE TYPES IS GREATE
     1R THAN 1 - RUN TERMINATED)
      GO TO 5
  950 PRINT 955
  955 FORMAT (1H0,2X,72HNO BALANCE AS TIME REQUIRED IS LESS THAT CRITICA
     1L  PATH TERMINATION DATE)
      GO TO 5
  960 PRINT 965
  965 FORMAT (1H0,2X,78HA RUN CANNOT BE MADE AS THE NUMBER OF ACTIVITIES
     1 IS LESS THAN A MINIMUM THREE.)
      GO TO 5
      END
      SUBROUTINE CRITIC
      DIMENSION EF(100), LS(100), LF(100)
      COMMON T(100), H(100), DUR(100), ES(100), TF(100), FF(100), CPTD,
     1SNORM, NACT, KEXIT, ASST(100), ASFT(100), RN(100,20), NRES
      INTEGER TACT, T, H, TF, FF, ES, DUR, RN
      REAL LS, LF
C ***  CPM SUBROUTINE  TO FIND OUT CRITICAL PATH
C
C *** TACT=TOTAL NUMBER OF ACTIVITIES.
C *** SNORM=START TIME FOR PROJECT.
C *** T(I)=TAIL NODE NUMBER FOR ACTIVITY I.
C *** H(I)=HEAD NODE NUMBER;H(I)>T(I).
C *** DUR(I)=DURATION TIME OF ACTIVITY I.
C *** CPTD=CRITICAL PATH TIME
C *** ES(I)=EARLIEST START TIME FOR ACTIVITY I.
C *** EF(I)=EARLIEST FINISH TIME FOR I.
C *** LS(I)=LATEST START TIME FOR I.
C *** LF(I)=LATEST FINISH TIME FOR I.
C *** TF(I)=TOTAL FLOAT FOR ACTIVITY I.
C *** FF(I)=FREE FLOAT FOR I.
C *** ASST(I)=SCHEDULE START FOR ACTIVITY I.
C *** ASFT(I)=SCHEDULE FINISH FOR ACTIVITY I.
C
C
C *** CHECK THE INPUT DATA
C
      TACT=NACT
      DO 5 I=1,TACT
      IF (H(I).LE.T(I)) GO TO 125
```

```
    5 CONTINUE
C
C *** PRIMARY SORTING ON TAIL NODES.
C
      MX=TACT-1
      DO 25 N=1,MX
      NX=N+1
   10 IF (T(N).LE.T(NX)) GO TO 20
      DO 15 L=1,NRES
      ITEMP=RM(NX,L)
      RM(NX,L)=RM(N,L)
   15 RM(N,L)=ITEMP
      JT=T(NX)
      JA=H(NX)
      JB=DUR(NX)
      AC=ASST(NX)
      AD=ASFT(NX)
      T(NX)=T(N)
      H(NX)=H(N)
      DUR(NX)=DUR(N)
      ASST(NX)=ASST(N)
      ASFT(NX)=ASFT(N)
      T(N)=JT
      H(N)=JA
      DUR(N)=JB
      ASST(N)=AC
      ASFT(N)=AD
   20 NX=NX+1
      IF (NX.GT.TACT) GO TO 25
      GO TO 10
   25 CONTINUE
C
C *** SECONDARY SORTING ON HEAD NODE.
C
      KX=TACT-1
      DO 50 K=1,KX
      KR=K+1
   30 IF (T(K).NE.T(KR)) GO TO 50
      IF (H(K).GT.H(KR)) GO TO 35
      GO TO 45
   35 JX=H(K)
      JY=DUR(K)
      AX=ASST(K)
      AY=ASFT(K)
      H(K)=H(KR)
      DUR(K)=DUR(KR)
      ASST(K)=ASST(KR)
      ASFT(K)=ASFT(KR)
      H(KR)=JX
      DUR(KR)=JY
      ASST(KR)=AX
      ASFT(KR)=AY
      DO 40 L=1,NRES
      ITEMP=RM(K,L)
      RM(K,L)=RM(KR,L)
   40 RM(KR,L)=ITEMP
   45 KR=KR+1
      IF (KR-TACT) 30,30,50
   50 CONTINUE
C
C *** CRITICAL PATH CALCULATIONS: FORWARD PASS.
C
      ES(1)=SNORM
      EF(1)=ES(1)+DUR(1)
      N=1
      NX=N+1
   55 IF (T(NX).NE.T(N)) GO TO 60
      ES(NX)=ES(1)
```

```
      EF(NX)=ES(NX)+DUR(NX)
      NX=NX+1
      GO TO 55
   60 CONTINUE
      DO 75 N=NX,TACT
      XB=0.0
      NA=T(N)
      DO 70 I=1,TACT
      IF (H(I).NE.NA) GO TO 70
      IF (EF(I).LT.XB) GO TO 65
      XB=EF(I)
   65 ES(N)=XB
   70 CONTINUE
      EF(N)=ES(N)+DUR(N)
   75 CONTINUE
C *** CRITICAL PATH LENGTH
C
      KA=H(TACT)
      RB=0.0
      DO 85 K=1,TACT
      IF (H(K).NE.KA) GO TO 85
      IF (EF(K).LT.RB) GO TO 80
      RB=EF(K)
   80 CPTD=RB
   85 CONTINUE
C
C *** FINDING OUT TOTAL FLOAT,FREE  FLOAT BY BACKWARD PASS.
C
      LF(TACT)=CPTD
      LS(TACT)=CPTD-DUR(TACT)
      NT=TACT-1
      NX=TACT
      DO 100 N=1,NT
      NX=NX-1
      IF (H(NX).NE.H(TACT)) GO TO 90
      LF(NX)=CPTD
      LS(NX)=CPTD-DUR(NX)
      GO TO 100
   90 XL=9999999.
      KR=NX+1
      DO 95 K=KR,TACT
      IF (T(K).NE.H(NX)) GO TO 95
      IF (LS(K).GT.XL) GO TO 95
      XL=LS(K)
   95 CONTINUE
      LF(NX)=XL
      LS(NX)=LF(NX)-DUR(NX)
  100 CONTINUE
C
C *** FINAL CALCULATIONS
C
      DO 120 J=1,TACT
      TF(J)=LF(J)-EF(J)
      JH=H(J)
      DO 105 K=J,TACT
      IF (JH.EQ.H(TACT)) GO TO 110
      IF (T(K).NE.JH) GO TO 105
      ESX=ES(K)
      GO TO 115
  105 CONTINUE
  110 ESX=CPTD
  115 FF(J)=ESX-EF(J)
  120 CONTINUE
      RETURN
  125 WRITE (6,130) T(I),H(I),I
  130 FORMAT (10X,48HERROR: HEAD NODE EQUAL OR GREATER THAN TAIL NODE,10
     1H T(I) IS  ,I2,13H AND H(I) IS ,I2,10H WITH I = ,I2,/)
      CALL EXIT
      END
```